Interacting with users in social networks: The Follow-back Problem

by

Krishnan Rajagopalan

B.S, U.S. Naval Academy (2014)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .....
Sloan School of Management
May 2, 2016

Certified by .....
Dr. Danelle Shah
Technical Staff
MIT Lincoln Laboratory
Thesis Supervisor

Certified by .....
Prof. Tauhid R. Zaman
KDD Career Development Professor in Communications and Technology
Assistant Professor of Operations Management
Thesis Supervisor

Accepted by .....
Prof. Dimitris Bertsimas
Boeing Professor of Operations Research
CoDirector, Operations Research Center

Interacting with users in social networks: The Follow-back Problem

by

Krishnan Rajagopalan

Submitted to the Sloan School of Management
on May 2, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

Imagine in a social network that an agent wants to form a connection with a target user(s) by interacting with the friends of the target(s). Because forming a connection is known as following in social networks such as Twitter, we refer to this as the *follow-back problem*. The friends of the target user(s) form a directed graph which we refer to as the "friends graph." The agent's goal is to get the target to follow him, and he is allowed to interact with the target and the target's friends. To understand what features impact the probability of an interaction resulting in a follow-back, we conduct an empirical analysis of several thousand interactions in Twitter. We build a model of the follow-back probabilities based upon this analysis which incorporates features such as the friend and follower count of the target and the neighborhood overlap of the target with the agent. We then use this model to solve the follow-back problem. We find optimal policies for simple network topologies such as directed acyclic graphs and single cycle graphs. For arbitrary directed graphs we develop heuristics based upon a graph score measure we define as the *followback score*. We show through simulation that these heuristic policies perform well on real Twitter graphs.

Thesis Supervisor: Dr. Danelle Shah
Title: Technical Staff
MIT Lincoln Laboratory

Thesis Supervisor: Prof. Tauhid R. Zaman
Title: KDD Career Development Professor in Communications and Technology
Assistant Professor of Operations Management

Acknowledgments

I would first like to thank three sponsoring institutions. I am grateful that the US Navy has given me the privilege of pursuing a Master's degree immediately upon my commissioning. I am grateful that it has continued investing in my development, and I look forward to repaying that debt in service during the coming years. I also owe thanks to MIT Lincoln Laboratory, and specifically, Group 104, which has sponsored me on a military fellowship. I'd also like to thank the MIT Operations Research Center, whose unparalleled students, faculty, and staff have helped make my experience at MIT so enriching.

Next, I'd like to thank two research mentors that have contributed to this thesis as well as my intellectual growth while at MIT. Dr. Tauhid Zaman of the MIT Sloan School of Management's Operations Management Group served as a dedicated and patient research adviser over the past two years. The energy and time he devoted to our research as well as his fun demeanor made learning from him a pleasure. Dr. Danelle Shah of MIT Lincoln Laboratory gave me tremendous support in my research and was always willing to offer a guiding hand. Her attention to detail as well as her emphasis on precision and clarity of presentation are qualities I will strive to emulate in the future.

Lastly, I would not have had such a great experience at MIT without the love and care of my family and friends. I'd like to thank my wife, Caitlin, who I got to spend the last two years growing with in the Boston area. I would like to also thank my parents and siblings for their love and support as well as their confidence in me.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of MIT Lincoln Laboratory, the United States Navy, Department of Defense, or the U.S. Government.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Business Motivation

Social networks have rapidly risen in size over the past decade and today represent one of the main platforms through which social interactions occur. Current estimates are that 74% of adult Internet users are on social media and in many other parts of the world, including emerging markets, this penetration rate is even higher [28]. Today people use social networks as the primary mechanism to interact with friends, make business connections, and find like-minded individuals. Therefore, it is important to understand how to effectively engage with users in a social network.

As a specific situation, consider an individual or company trying to establish its presence in a new market. One way to build its brand recognition is to advertise over social media. Presumably, there is a group of influential individuals in this market that already have well established social media presences. The company that is new to the market may want to connect with these influential users to tap into their well-established networks. What is the best way to do this over social media? In the off-line setting, humans understand the basics of social interaction. For instance, it is harder to connect with very popular people versus less popular people. Also, it is easier to connect with someone if you share mutual contacts. We wish to understand if these social intuitions translate into the online setting and if so, how does one implement these intuitions? Such an understanding would allow for more effective

interactions in social networks. People could more easily connect with users they deem interesting. Companies and organizations could engage with more customers and grow their online social networks more quickly. Individuals could systematically gain influence and followers in social networks. In short, if we understand what drives online interactions, we can develop policies to make us more effective users of social networks.

1.2 National Defense Motivation

With its rapid growth, social media has gained the interst of national defense policy makers and strategists. In January of 2016, The White House held a summit with executives from some of the leading American technology firms including Facebook and Twitter to help generate strategies to counter extremist recruitment online [34]. Defense Secretary Ash Carter met with officials from Facebook in April of 2015 [3].

Anti-Access/Area Denial (A2/AD) is a defense paradigm used to describe a strategy of preventing one's adversary from operating military forces within a specific region of interest. United States defense strategists consider the A2/AD capability posed by American adversaries to be one of the greatest threats of the twenty-first century [29]. An adversary can also pose an A2/AD threat over the information space, which encompasses both physical and non-physical domains where information is spread [29]. This is a threat that the United States has increasingly seen.

The Chinese Communist Party has recognized the value of popular content on social media. It employs thousands of social media operators, or "wumao dang," to share material directed by the state [21]. These accounts help draw attention to topics the party wants to highlight [21]. By intentionally inundating the social media platform, the tool can hinder the free movement of alternative ideas and information and thus can be analyzed through the A2/AD paradigm. Non-state adversaries have employed similar strategies.

In summer 2014, the Islamic State in Iraq and the Levant (ISIL) launched an application on the Google Play Store called "The Dawn of Glad Tidings." If a Twitter

user downloaded this application, his account could be used by ISIL to post content [6]. The idea behind the application was simple: use existing Twitter accounts of jihadist-sympathizers to amplify the message of the terror group. However, the consequences were far greater. By hijacking Twitter accounts to broadcast its message, ISIL built a quasi-active community of supporters on the social media site. An interested onlooker would perceive a network of Twitter users engaging one another and posting content sympathetic to the jihadist group's cause, even while much of this activity was automated. ISIL social media recruiters also deliberately attempt to isolate their targets from the rest of the world by encouraging them to terminate their existing social media connections [9]. ISIL attempted to restrict the flow of alternative information to its target audience: it **denied** its prospective recruits free **access** to information. ISIL recruits by framing a perverse, inflammatory narrative about the tenets of Islam and the West's subjugation of the Muslim world [6]. It can recruit more easily where opposing voices are absent. It relies, in part, upon an A2/AD capability to block counter-propaganda efforts.

The United States Department of Defense uses Military Information Support (MIS) units to penetrate contested information environments. Military Information Support Operations (MISO) are the conveying of specific information to target foreign audiences in order to sway their perceptions and ultimately influence their actions in support of national objectives [1]. As social media platforms have made the spread of information more nuanced, MIS units require tools that will allow them to penetrate this new, increasingly important space, in the event that an adversary employs an A2/AD capability. When an adversary is attempting to use propaganda to influence a target audience over social media, MIS units may be tasked with inserting counter-propaganda information in the network.

One problem confronting MIS units is ensuring that their counter-propaganda efforts can penetrate an information blockade over social media. On Twitter and other social media platforms, for instance a user will generally only see the content posted by users who he "follows" or is connected with. MIS units can improve their ability to penetrate contested social media platforms by learning how to form desired

connections. We study strategies that can be used by an agent when it desires to form a connection with a target user or users and call this the *follow-back problem*.

The author does not advocate that the US Government come even remotely close to mirroring the tactics employed by our adversaries. The United States should never compromise our commitment to free press. However, US policymakers must be realistic in recognizing that all of our potential adversaries might not conform to this value. If and when one of these adversaries uses an A2/AD capability against us in the rapidly changing information domain, we must be able to penetrate the blockade and deny them any tactical advantage.

1.3 Our Contribution

We study the the *follow-back problem*, where the goal is to find the most effective way to interact with users in a social network in order to form connections with them. The name is based on terminology from the social network Twitter, where when a user forms such a connection he is said to be a follower. We begin by formally defining the follow-back problem in mathematical terms. The problem definition assumes some model for user behavior in social networks, in particular, what features affect the probability of being followed. To better understand this model, we conduct an online field experiment in Twitter where online agents interact with users in order to gain followers. Our experimental findings show that the follow probability is affected by the number of friends and followers of the targeted user and also the number of friends of the target user that follow the agent, a feature we refer to as the *overlap*. In particular, we find that the follow probability is larger if the targeted user has many friends, few followers, and a large overlap with the agent. These observations match our social intuition which suggets that it is easier to connect with people who are less popular (many friends, few followers) and who have mutual contacts (large overlap).

We propose a simple model for the follow probability based on this empirical analysis and use this model in our analysis of the follow-back problem. We are able to provide the optimal interaction policy for directed acyclic graphs (DAGs) and

provide an approximation for the expected number of follows this policy gives on a DAG, which we define as the DAG's *follow-back score*. We use follow-back score to design two different algorithms that, given an arbitrary directed graph, try to find a good candidate directed acyclic subgraph. We show that our heuristics perform well on different real and synthetic social media graphs.

The thesis is outlined as follows. We review related literature in 1.4 and formally define the follow-back problem in 1.5 of this chapter. We then present our empirical analysis of follow probabilities in the following chapter 2. We analyze optimal policies for the follow-back problem and define follow-back score in chapter 3. Next, we discuss our algorithm for a graph in which there is a single target and present results for this algorithm in 4. Then, we introduce the general follow-back problem for a graph with multiple targets and show how constrained optimization is used to find a policy in 5. We explain some avenues of future work and conclude in 6.

In this article, we propose the follow-back algorithm to maximize the likelihood of being connecting with, or being "followed," by selected user(s) on the social network Twitter. The main contributions of our work:

- We believe that our research is the first to formalize a problem of connecting with a desired user or group of users over social media.
- We run field experiments on Twitter and present features of a user that impact his propensity to connect, or "follow."
- We use the insights from our experiments to provide algorithms that social media operators can use to increase the chances of gaining desired connections.

1.4 Literature Review

There are several lines of research related to our work. First, there is work on empirical studies of influence in social networks. Second, there is the body of work on the sociological notion of triadic closure and its manifestation in online social networks. Third, there is a sizable body of work on theoretical analysis of influence maximization

in social networks. Fourth, there is a well-studied problem called the maximum acyclic subgraph problem. Finally, there is the broad area of network centrality measures.

Many researchers have studied what factors affect influence in social networks. Influence can be roughly defined as the ability to cause an individual to adopt a new product or behavior. The ability of social influence to affect health behavior was demonstrated in [14] and [15]. The effect of the local structure of a social network on health behavior was demonstrated in [10]. The use of randomized online experiments to discern causal social influence was done in [4]. In [5] a massive randomized online experiment in Facebook was carried out to infer what types of users are influential and susceptible. It was found that influence and susceptibility depended on demographic characteristics as well as network structure. [27] analyzes the Twitter network structure as well as the temporal aspects and network structure of retweets. [11] analyzes the influence of users in a network based upon number of retweets and number of followers. [22] showed that data from Facebook could be used to understand structural features of networks to better understand social contagion. [5] used Facebook data to try and infer which users are classified as influential and which are classified as susceptible to being influenced based upon demographic characteristics as well as network structure. [4] showed that using data made from randomized interactions in social media are valuable in understanding influence.

The overlap feature which we present in 2.5 can be considered as a directed graph version of the notion of triadic closure from sociology, or "directed closure" [16]. A graph theoretic view of triadic closure is provided in [17]. This principle is used in [8] to design a social botnet on Facebook. The algorithm in [8] strives to infiltrate a Facebook network by leveraging the principle of triadic closure. They initially acquire friends from a set of vulnerable users and then snowball their engagements by identifying users with whom they have mutual connections. The effect of social overlap on influence was shown in [22]. Here a massive Facebook dataset was studied and it was found that adoption of Facebook by a new user depended strongly on how many people invited that user and their local network structure. A similar result is found in [18] where it is shown that the likelihood of accepting a connection request

in an online social network is about three times higher given the existence of some number of mutual connections.

Related to the follow-back problem is the influence maximization problem where the goal is to maximize the spread of a piece of information through a social network by seeding a select set of users. One of the first theoretical studies of the influence maximization problem was in [24] where it was shown that the problem was sub-modular, so a greedy algorithm would have good performance. This work led to subsequent variations of the problem and different algorithmic solutions [25], [12], [13]. Each of these works exploited the sub-modular structure of the problem to obtain efficient algorithms.

The study of Directed Acyclic Graphs (DAGs) overlaps with the study of partially ordered sets, since a partial order may be imposed on the nodes of a DAG. [33] discusses the process of "refining" a partial order. Finding a directed acyclic subgraph for an arbitrary directed graph that is the most refined partial order on the graph possible is deemed the maximum acyclic subgraph problem. It seeks to select the maximum number of edges from the directed graph subject to the induced subgraph being acyclic [35]. This problem is generally NP-hard.

Many problems in graphs can be solved through the development of functions which map vertices to real numbers. These functions are known as network centralities. They quantify how central a vertex is to the problem at hand, with the definition of centrality being set by the problem at hand. In our work we use centralities to both compare the performance of our algorithms as well as to supplement our algorithms themselves. One of the simplest is distance centrality which measures how close a vertex is to all other vertices [31]. A related measure is betweenness centrality, which measures how many paths pass through a vertex [19], [20]. This measure gives importance to vertices that are critical to the flow through a graph. Eigenvector centrality measures how important a vertex is by how important its neighbors are [7]. This centrality measure is closely related to the PageRank [30] and HITS [26] centrality measures. Rumor centrality was developed to find the source of a rumor that spreads in a graph [32]. It was shown that the vertex with the highest rumor

centrality corresponded to the maximum likelihood estimate of the rumor source in some specific graph topologies and that rumor centrality was related to the number of linear extensions of a partially ordered set.

1.5 The Follow-back Problem

Initially at time $n = 0$ we have a primary target vertex t, a set of vertices t follows, which we refer to as the friends of t, and an agent vertex a. The friends form the friends graph $G_0 = (V, E_0)$ where E_0 is the initial edge set. A directed edge (u, v) pointing from u to v for some $u \neq v$ in the friends graph means that u is followed by v. In a social network such as Twitter this means that any content u posts will be visible to v. V is the fixed vertex set of the graph, which we define as including the friends, t, and a. Each vertex v has a set of features F_v associated with it and we define the set of all vertex features as $\mathbf{F} = \bigcup_{v \in V} F_v$. For the friends we have that $(v, t) \in E_0 \ \forall v \in V \backslash a$. This is because initially t follows all vertices in V except for a. We will assume that a initially has no edge with any vertex in V. The goal in the follow-back problem is for the agent a to have the primary target t follow him by interacting with the vertices in V. In other words, a wants to create the edge (a, t). We illustrate the setting of the follow-back problem in Figure 1-1.

At each time n, a can interact with a single vertex in V. The vertex chosen by a at time n is denoted u_n. The interaction of the agent with a target vertex is some form of social network interaction. In Twitter these interactions include following, retweeting, and mentioning. We require that $u_n \neq u_i$ for any $i < n$, meaning that a can only interact with each $v \in V$ once. We define the interaction policy of a as the vertex sequence $\pi = \{u_1, u_2, ...\}$ along with the sequence of interactions $x = \{x_{u_1}, x_{u_2}, ...\}$. Here x_{u_i} specifies the type of Twitter interaction (following, retweeting, or mentioning) with u_i. When at time n the agent engages with u_n, either u_n will follow a or u_n will not follow a. We let the random variable X_{u_n} be one if the engagement of a results in u_n following a, otherwise X_{u_n} is zero. The vertex set does not change, but the actions of a can change the edge set E_n. If the agent's action x_{u_n} gains a follow from vertex u_n,

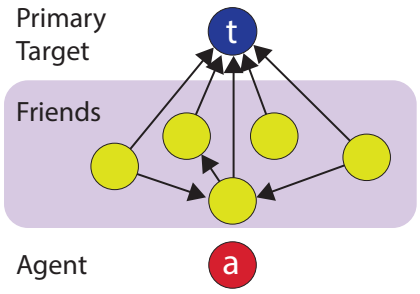

Figure 1-1: Illustration of the setting of the follow-back problem. The agent a wants to be followed by the primary target t by interacting with the vertices in the friends graph.

then we update the graph to $G_n = (V, E_n)$ where $E_n = E_{n-1} \bigcup (a, u_n)$. Furthermore, we assume that a target cannot follow the agent without being interacted with first and that once the target follows the agent, he cannot "unfollow." This means that edges can only be formed in response to an interaction by the agent and edges cannot be destroyed.

The state at time n is the graph G_n and the vertex feature set $\mathbf{F}$. Denote the follow probability of a vertex u_n in response to the agent interaction x_{u_n} as $\mathbf{P}\left(X_{u_n} = 1 | G_n, x_{u_n}, \mathbf{F}\right)$. We are assuming here that the follow probability of u_n only depends upon the current state of the graph, the type of interaction, and some fixed vertex features.

In the follow-back problem, the agent selects the policy (π, x) to maximize the probability that the target t will follow back. That is, the agent chooses a policy (π, x) to solve the following problem:

$$\max_{(\pi,x)} \mathbf{E}\left[X_t | \pi, x, G_0, \mathbf{F}\right]. \tag{1.1}$$

We will see that the solution to this problem depends upon understanding the structure of the follow probabilities which characterize the interaction dynamics. Our empirical analysis in Section 2 will show how these follow probabilities depend upon various features of the vertices and the friends graph.

Chapter 2

Empirical Analysis of Interactions in Social Networks

To determine the structure of the follow probabilities we require data on interactions in social networks. To obtain such data we conducted an online experiment involving user accounts to which we were given access. The accounts belonged to six Moroccan artists who were interested in connecting with Twitter users interested in Moroccan art. The artist accounts would interact with Twitter users using different types of interactions, and we recorded the results of the interactions after one week. This was to allow time for the reaction of the targets to occur, as many times they may not be immediate. We were able to obtain data for over 6,000 Twitter interactions. We now discuss our empirical findings in detail.

2.1 Empirical Setup

We conducted a series of experiments to understand different aspects of interactions in social networks. As mentioned above, we utilized six different Twitter accounts from Moroccan artists. These accounts acted as our agents. We conducted three different experiments. The first experiment was designed to understand the effect of the different types of interactions in Twitter on the follow probability. The second experiment aimed to understand how vertex features such as friend count and follower

count affect the follow probabilities. The third experiment looked at more complex graph features, such as the number of the agent's followers who are also friends of the target, which we refer to as the *overlap*.

The empirical procedure was as follows. For the first two experiments we had the agents use the Twitter search API to find target users who have posted tweets about Morocco or art. The agents then interacted with all of these targets in some way. We had the agents perform this search and interact procedure in randomly spaced intervals that were on average two hours apart. The experiments were each run for one week. We waited one week from when the interaction occurred and checked if there was any sort of reaction from the targets. For the third experiment the procedure was the same, except to find target users, in addition to using the search API, we also searched for followers of users who followed-back the agents during the first two experiments.

We made sure that each target user only interacted with a single Moroccan artist account to avoid the risk of a target's reaction being confounded by multiple interactions. Also, each target user was only interacted with one time. The six different Twitter agent accounts we used have different names and profile images, but the content they posted on their Twitter Timelines is similar. This content consisted of images of their artwork, which was similar across the different accounts.

We modeled the follow probabilities using a logistic regression model. If we denote the follow probability of a target vertex v as $\mathbf{P}\left(X_v = 1 | G, x, \mathbf{F}\right) = p(S_v)$ where S_v is the set of all data associated with the interaction. Under our logistic regression model the follow probability for a vertex v is

$$\log\left(\frac{p(S_v)}{1-p(S_v)}\right) = \sum_{k=1}^{d} \beta_k S_{vk}, \tag{2.1}$$

where we have assumed that for a vertex v the set S_v has d elements.

2.2 Interaction Type

There are three main ways an agent can interact with a target in Twitter. The first is following. When the agent follows a target any content the target posts is shown to the agent in his Twitter timeline. In addition, when the agent follows the target, the target is notified that the agent is now following him. The second interaction is retweeting, which is reposting one of the target's tweets. The agent's timeline displays the original tweet, and the target is notified of the retweet. The third interaction is replying, which involves the agent creating a tweet that mentions the target. The tweet appears on the agent's timeline and the target is notified.

We had one agent account post original content and not interact with other users. This agent acted as a control. The remaining five agents interacted with 150 to 220 different targets over a one week period. Each agent was assigned a specific type of interaction. These interactions were following, retweeting, replying, retweeting and following, and replying and following. Agents who replied used a fixed set of messages such as "I like that" or "Nice".

The target can react to the agent's interaction by mentioning, retweeting, or following. If the agent's interaction produces a reaction from the target, we refer to this as a conversion. We show the conversion rate for each combination of agent interaction and target reaction in Figure 2-1. The control agent that did not interact with anyone is referred to as "None" in the figure. This agent did not receive any form of target reaction. One observation from this figure is that to gain followers, the agent must interact with others.

We run our regression model for the target reaction of following using three indicator variables for following, retweeting, or replying. Table 2.1 shows that the only interaction type that was significant (at the 5% level) in predicting the follow reaction was following, which increased the follow probability. This analysis shows that for the follow-back problem, the best interaction to use is following.

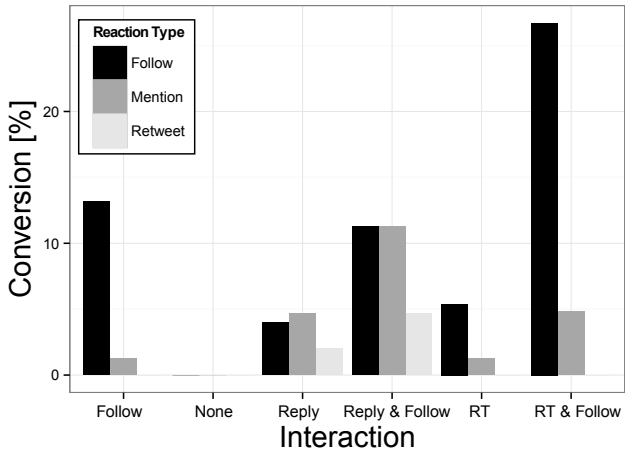

Figure 2-1: Conversion rate for different agent interaction and target reaction combinations (RT stands for retweet).

Table 2.1: Interaction Type

Feature	β	p-value
Intercept	-4.18***	3.25×10^{-15}
Follow	2.14***	1.25×10^{-5}
Retweet	-0.14	0.68
Reply	-0.22	0.48

The significance codes are *** : 0.001, ** : 0.01, and * : 0.05.

2.3 Interaction Type, Follower Counts, Friend Counts

We next run a regression using the same data set from above but include the agent and target friend and follower counts. We see that in table 2.2, after adding friend and follower counts to the regression, the significance of the interaction types decrease, as the target friend and follower counts are the features with highest significance. This finding implies that social media interactions may be too complicated of phenomena to adequately predict responses with simple indicator variables. People respond very differently to the varying forms of interaction available over social media–both verbal and non-verbal. However, the fact that the agent who did not initiate interaction with anyone gained no following, supports the intuitive idea that interacting in some manner is best if one wants to gain connections. Being an isolated user that avoids interaction does not lend well to establishing a network. In practice, to most effectively gain followers, a social media operator would want to tailor each interaction so that

Table 2.2: Interaction Type

Feature	β	p-value
Intercept	-10.6994***	1.31×10^{-08}
log10(target friend count)	3.4066***	6.28×10^{-10}
log10(target follower count)	-1.9575***	1.03×10^{-05}
log10(agent friend count)	0.2954	0.7634
log10(agent follower count)	1.4605	0.2883
Follow	1.6365*	0.0211
Retweet	-0.7689	0.1096
Reply	-0.9330*	0.0238

The significance codes are *** : 0.001, ** : 0.01, and * : 0.05.

it has the greatest chance of winning the favor of the target. We further investigate the significance of follower and friend counts in 2.4.

2.4 Friend and Follower Counts

Our next experiment aimed to understand how the friend and follower counts of both the target and agent impacted the follow probability. Based on our study of different Twitter interactions, we decided to have the agents only follow and retweet the targets. Prior to this experiment, the agents obtained followers from an online service. This allowed for sufficient variance in their follower counts, with one agent receiving as many as 10,000 followers. These follower accounts appeared to be automated bot accounts. However, our interest was in changing the follower count displayed on the agent's profile, not in the quality of the followers themselves. We wanted to see if having a higher follower count would make the agent seem more important and increase the follow probability. We were able to measure over 2,000 interactions for the agents.

We first examine the impact of the agent follower count on the follow probability. In Figure 2-2 we plot the conversion rate for follows versus the follower count of the agent. The results indicate that the follower count of the agent does not have a significant impact on the conversion rate for follower counts up to 10,000. This is somewhat counter intuitive as we would expect users to be more likely to follow

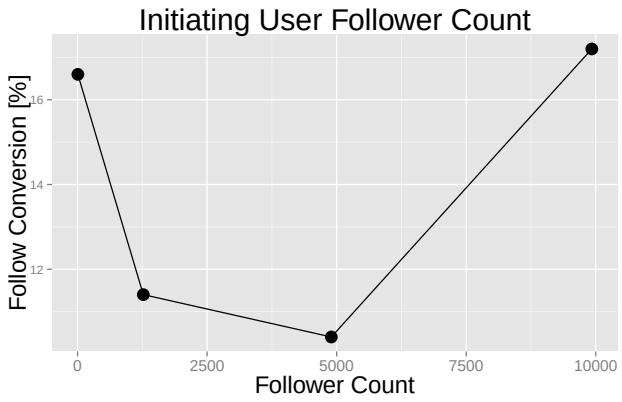

Figure 2-2: Plot of the follow conversion rate versus agent follower count. The horizontal axis shows the follower counts of four agents. The vertical axis shows the percentage of targets that followed each agent.

someone with a higher follower count, as this is a symbol of status on Twitter. Our data suggests that it is not clear if the agent's follower count has any effect on the follow probability.

The follower and friend counts of the target had a clearer relationship to the follow probability. We plot in Figure 2-3 the friend and follower count of each target. We use different markers for those that followed the agent and those that did not. A clear pattern is seen here. Target's with lower follower counts and higher friend counts tended to follow the agents more frequently. This makes sense intuitively as we would assume that user's with higher friend-to-follower ratios have a propensity to follow others on Twitter.

Our regression model used the logarithm (base 10) of the friend and follower counts of the agent and targets as features. The results of our model fitting is shown in Table 2.3 and support what was seen in the figures. The follower and friend counts for the target were significant while the friend and follower counts of the agents were not. Also, targets with many friends and few followers were more likely to follow the agent.

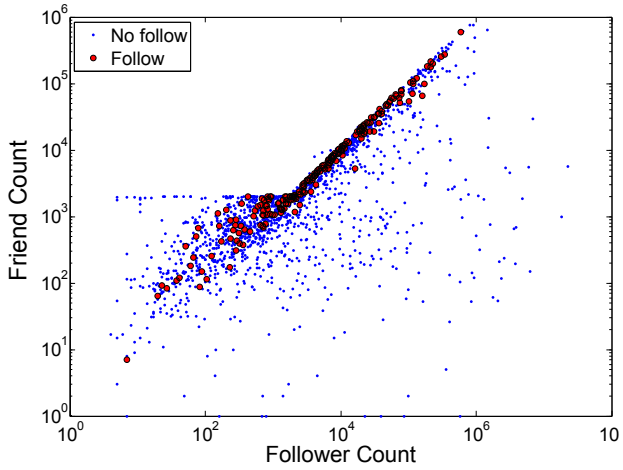

Figure 2-3: Plot of the follower count versus the friend count for each target. The markers indicate which targets followed the agent and which did not.

Table 2.3: Follower and Friend Counts

Feature	β	p-value
Intercept	-2.641***	$< 2 \times 10^{-16}$
$\log_{10}$(Target Friend Count)	3.41***	6.28×10^{-10}
$\log_{10}$(Target Follower Count)	-2.0***	1.03×10^{-5}
$\log_{10}$(Agent Follower Count)	1.5	0.29
$\log_{10}$(Agent Friend Count)	0.30	0.76

The significance codes are *** : 0.001, ** : 0.01, and * : 0.05.

2.5 Overlap

Our final experiment looked at the overlap feature, which is related to the concept of triadic closure from social network theory. Triadic closure is the property that if two individuals A and B both have strong ties to C, then it is likely that some form of tie exists between A and B. Strong triadic closure states that, given this mutual connection with C, a weak tie between A and B *must* exist. Triadic closure is a term usually applied to undirected graphs. [16] posits the existence of an analog in a directed network, or "directed closure." We define the *overlap* for two vertices. Formally, the overlap of a vertex v with another vertex u is equal to the number of directed two-hop paths from v to u. In Twitter terms, user v has an overlap with user u equal to the number of friends of u that follow v. We illustrate this overlap feature in Figure 2-4. The intuition behind this definition is based on the flow of information.

In Twitter content posted by a user can be seen by his followers. Therefore, the target could see some of the agent's content if it is retweeted by friends of the target that follow the agent. The target could also see any replies these users give to the agent. Using our terminology, we call these users members of the agent's *overlap* with the target. Seeing this content may make the target more interested in the agent and result in an increased follow probability.

We had the agents select targets by using the Twitter search API and also by choosing the followers of users who had previously followed-back the agent. This way we were able to obtain targets with zero and positive overlap. The agents retweeted and followed each target. We plot the follow conversion rate versus the overlap in Figure 2-5. As can be seen, the conversion rate increases as the overlap increases. The increase is not monotonic, but this may be due to not having sufficient data points for different overlap values, as seen by the 95% confidence intervals in the figure. Our regression model uses overlap as a feature along with the log transformed friend and follower counts of the target and the results are shown in Table 2.4. As can be seen, the overlap is a significant feature that increases the follow probability.

Another interesting result is found if we compare our measured conversion rates with those from [22]. In that work, the conversion rate was for users who were invited to Facebook that actually joined, and the overlap was the number of email invitations the user received from distinct people. We plot conversion rate versus overlap from our Twitter data and the Facebook data from [22] in Figure 2-6. We normalize our results so that an overlap of one has conversion rate one, just as in [22]. Remarkably, we see that the two curves are almost exactly equal, with slight differences for an overlap of four. The normalized conversion rate is the same for gaining a follow in Twitter and joining Facebook. These are very different actions in terms of effort required, and the overlap has a different meaning in each situation. This suggests that the overlap feature may be an important feature in a wide variety of social networks and its impact may be very similar.

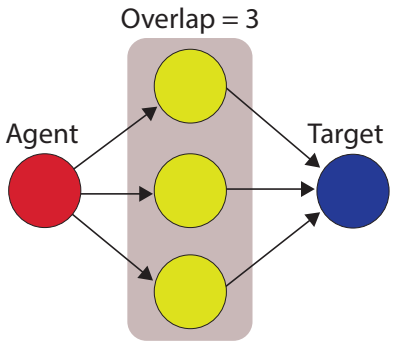

Figure 2-4: Illustration of the overlap feature between the agent and target vertex. The overlap in the figure is three.

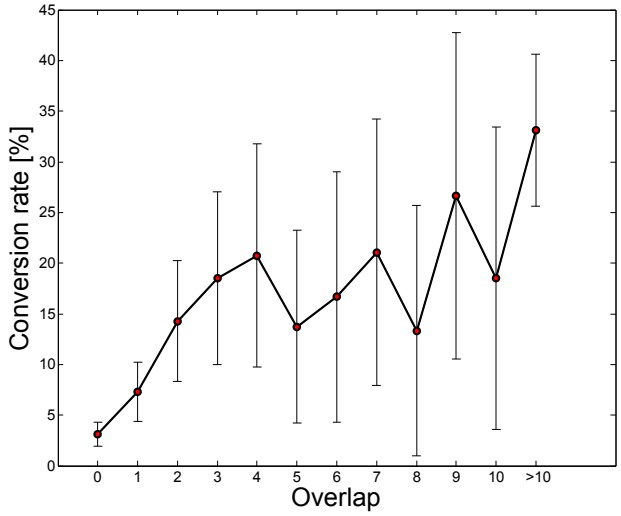

Figure 2-5: Plot of the follow conversion rate versus overlap. The red circles are the median and the error bars are 95% confidence intervals.

Table 2.4: Overlap

Feature	β	p-value
Intercept	-2.49***	4.10×10^{-8}
Overlap	0.28***	6.57×10^{-12}
$\log_{10}$(Target Friend Count)	0.45*	0.05
$\log_{10}$(Target Follower Count)	-0.63***	1.77×10^{-4}

The significance codes are *** : 0.001, ** : 0.01, and * : 0.05.

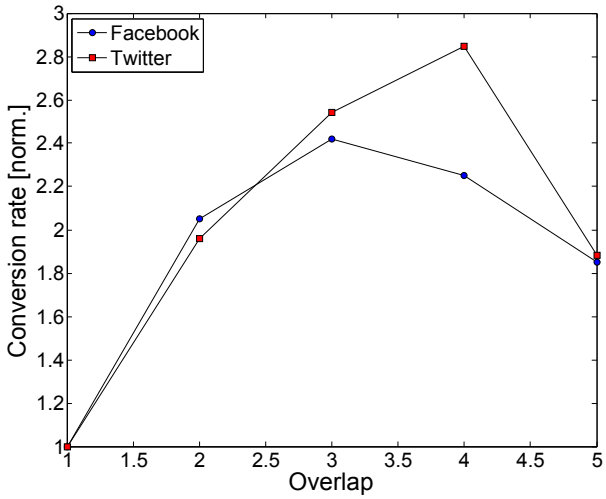

Figure 2-6: Plot of the normalized conversion rate for our data on follows in Twitter and Facebook signups from [22] versus overlap.

Chapter 3

Theoretical Development

3.1 Optimal Policies for the Follow-Back Problem

We now study optimal policies for the follow-back problem on different graph topologies. Recall that the goal here is to maximize the probability that a primary target vertex follows the agent by interacting with the primary target and all vertices it follows. The vertices following the primary target form the friends graph of the primary target. Recall that we saw in 2.3 that the type of interaction, i.e. follow, retweet, or reply, was insignificant in our follow probability regression, so we will consider only one type of generic agent interaction. As a result, the interaction portion x of a policy will be the same for all policies, and we can ignore it. Furthermore, for simplicity, we will assume that the agent's interaction(s) with one user must conclude before he begins interacting with another user. In other words, the agent cannot begin interacting with one user, start communicating with another user, and switch back to interacting with the former. In this case only the vertex sequence π determines the policy.

3.1.1 Follow Probability Model

To solve the follow-back problem we must make some general assumptions on the follow probabilities. Recall that earlier we assumed the follow probability depended

upon vertex features, graph state, and interaction type. Since we are assuming one type of interaction, we do not need to consider this in the functional form of the follow probability. Based on our empirical analysis, the only vertex features we need to consider are the friend and follower count of the target. For a target v we will call this set of features F_v. However, our proceeding analysis analysis applies to any general vertex features F_v. Also, as our empirical analysis showed, the graph feature we need to include is the overlap between the agent and target. We define the overlap between the agent and target v as ϕ_v.

With these assumptions we can write the follow probability of a target v in response to an interaction by the agent as $p(\phi_v, F_v)$. We further assume that the follow probability has the product form $p(\phi_v, F_v) = f(\phi_v)g(F_v)$. The product form assumption can be viewed as an approximation to the logistic form of the follow probability given by equation 2.1 when the probability value is small. Assume the coefficient of the overlap is β_ϕ and the coefficient vector of the vertex feature set is β_F. From our logistic regression model, the follow probability is given by

$$\begin{aligned} p(\phi_v, F_v) &= \frac{e^{\beta_0+\beta_\phi\phi_v+\beta_F^T F_v}}{1+e^{\beta_0+\beta_1\phi_v+\beta_F^T F_v}} \\ &\approx e^{\beta_0+\beta_\phi\phi_v} e^{\beta_F^T F_v} \\ &\triangleq f(\phi_v)g(F_v). \end{aligned}$$

The above expression is a good approximation when the term in the exponent is much less than zero, which is a valid assumption given our empirical observations. We use this approximation because it allows us to obtain simple analytic solutions for the follow-back problem on different graph topologies.

Finally, we assume that the follow probabilities are monotonically increasing in the value of the overlap, which we saw in figure 2-5. With this assumption, it is clear that an optimal policy for the follow-back problem is for the agent to first maximize its overlap with the primary target and then follow the primary target. Therefore, the follow-back problem reduces to choosing a sequence of vertices π in the friends graph to follow which maximizes the expected number of follows of the agent. For this

reason, we do not need to consider the edges of the primary target. In the following analysis, any assumptions on the friends graph topology refer to the graph with the primary target edges removed.

3.1.2 Optimal Policy on a Directed Acyclic Graph (DAG)

We begin by analyzing the follow-back problem on the simplest graph topology, a directed acyclic graph (DAG). The analysis is simple here because the graph contains no cycles. Because the follow probabilities are increasing in overlap, the intuitive solution here would be to maximize the overlap for each interaction by following a vertex after following its parents. To formalize this, recall that a DAG can also be viewed as a partially ordered set (poset) on the vertices, with the partial order requiring that a vertex must come after its parents. Any sequence of vertices which respects this partial order is known as a linear extension. For DAGs, we have the following result, which matches the intuition.

Theorem 3.1.1. *Let π^* be the optimal policy for the follow-back problem with initial graph G which is a directed acyclic graph. Then π^* is a linear extension of G.*

What is interesting about this result is that any linear extension is an optimal policy. This is because all linear extensions result in the same number of expected follows. Intuitively, this is true because the follow probabilities depend only upon features of the target and the overlap. Under our assumptions on the follow probabilities, interchanging the order of vertices in the policy in a way that respects the partial order does not affect any of these features. We illustrate the linear extension policy in Figure 3-1. Here there are multiple optimal policies corresponding to each linear extension of the graph. It does not matter which one is selected because they all respect the order relationships of the graph.

Proof. Let the friends graph be $G = (V, E)$ which is a DAG. Consider two vertices $u, v \in V$ such that there is a directed path from u to v in G and consequently no path from v to u because G is a DAG. We define two policies $\pi = \{u_1, ..., u, v, ..., u_n\}$

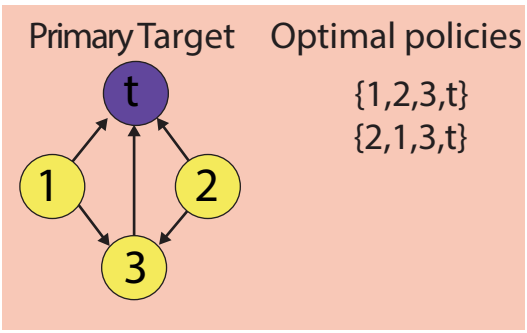

Figure 3-1: Illustration of several optimal linear extension policies on a DAG. The primary target is followed last for every optimal policy after the agent has interacted with every vertex in the friends graph.

and $\pi' = \{u_1, ..., v, u, ..., u_n\}$. The only difference in the two policies is that u and v are swapped. In π vertices u and v respect the partial order of G, whereas in π' they do not. We will now show that the expected follows of π' is less than π.

The expected follows of a policy π can be written as $\sum_{w \in V} \mathbf{E}\left[X_w | \pi\right]$, where X_w is one if the agent interaction with w results in a follow by w and zero otherwise. We want to show that the expected follows will be larger if u is followed before v. For any vertex w we define $\delta_w = \mathbf{E}\left[X_w | \pi\right] - \mathbf{E}\left[X_w | \pi'\right]$ and we define the difference in the expected follows of the two policies as $\delta = \sum_{w \in V} \delta_w$. For any vertex w which occurs before v and u in either policy, we have that $\delta_w = 0$. This is because the state of the graph when the agent interacts with these vertices is the same under each policy. For vertex u we have $\delta_u = 0$ because u does not follow v and there is no path from v to u. Therefore, there is no way for v to affect the overlap of u with the agent, and thus no way to affect the subsequent follow probability. For v and all vertices after u and v we use the following lemma.

Lemma 3.1.2. *Let $(u_1, u_2, ..., u_n)$ be a path on a directed graph G. Then $\mathbf{E}[X_{u_j} | X_{u_i} = 1] > \mathbf{E}[X_{u_j} | X_{u_i} = 0]$ for $1 \leq i < j \leq n$.*

This lemma shows that if the agent can get a vertex w to follow it, it will receive a boost in the expected follows for any vertex on a path from w. Now consider vertex

v. Using Lemma 3.1.2 we have

$$\begin{aligned}
\delta_v =& \mathbf{E}\left[X_v | X_u = 1, \pi\right] \mathbf{E}\left[X_u | \pi\right] \\
&+ \mathbf{E}\left[X_v | X_u = 0, \pi\right] \left(1 - \mathbf{E}\left[X_u | \pi\right]\right) \\
&- \mathbf{E}\left[X_v | X_u = 0, \pi'\right] \\
=& \mathbf{E}\left[X_u | \pi\right] \left(\mathbf{E}\left[X_v | X_u = 1, \pi\right] - \mathbf{E}\left[X_v | X_u = 0, \pi\right]\right) \\
>& 0.
\end{aligned}$$

Above we have used Lemma 3.1.2 along with the fact that X_i are Bernoulli random variables, so $\mathbf{E}[X_i] = \mathbf{P}(X_i = 1)$. We also made use of the fact that $\mathbf{E}\left[X_v | X_u = 0, \pi\right] = \mathbf{E}\left[X_v | X_u = 0, \pi'\right]$ because if u does not follow the agent, the follow probability is the same under both policies when the agent interacts with v.

For any vertex w that occurs after u and v we have two situations. Either there is no path from v to w, in which case $\delta_w = 0$ because v cannot affect the follow probability of w, or there is such a path. In the case where a path exists, we have

$$\begin{aligned}
\delta_w =& \mathbf{E}\left[X_w | X_v = 1, \pi\right] \mathbf{E}\left[X_v | \pi\right] \\
&+ \mathbf{E}\left[X_w | X_v = 0, \pi\right] \left(1 - \mathbf{E}\left[X_v | \pi\right]\right) \\
&- \mathbf{E}\left[X_w | X_v = 1, \pi'\right] \mathbf{E}\left[X_v | \pi'\right] \\
&- \mathbf{E}\left[X_w | X_v = 0, \pi'\right] \left(1 - \mathbf{E}\left[X_v | \pi'\right]\right) \\
=& \mathbf{E}\left[X_w | X_v = 1\right] \left(\mathbf{E}\left[X_v | \pi\right] - \mathbf{E}\left[X_v | \pi'\right]\right) \\
&+ \mathbf{E}\left[X_w | X_v = 0\right] \left(\mathbf{E}\left[X_v | \pi'\right] - \mathbf{E}\left[X_v | \pi\right]\right) \\
=& \left(\mathbf{E}\left[X_v | \pi\right] - \mathbf{E}\left[X_v | \pi'\right]\right)\left(\mathbf{E}\left[X_w | X_v = 1\right] - \mathbf{E}\left[X_w | X_v = 0\right]\right) \\
=& \delta_v \left(\mathbf{E}\left[X_w | X_v = 1\right] - \mathbf{E}\left[X_w | X_v = 0\right]\right) \\
>& 0.
\end{aligned}$$

Here, in addition to our result for δ_v, we also used the fact that the expected value of X_w conditioned on X_v is the same for both policies because the relevant graph state is the same. This result shows that $\delta > 0$, and π has more expected follows

than π'. Therefore, any policy can increase its expected follows by swapping any pair of adjacent vertices so they respect the partial order of the underlying DAG. This process can be continued until the policy is a linear extension of the DAG and no further increase in the expected follows is possible. □

of Lemma 3.1.2. We will prove the result by induction. For any $1 < i < n$ we have that $\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 1] > \mathbf{E}[X_{u_{i+1}}|X_{u_i} = 0]$ because the follow probabilities are monotonically increasing in the overlap and u_{i+1} follows u_i. This gives our base case of $\mathbf{E}[X_{u_2}|X_{u_1} = 1] > \mathbf{E}[X_{u_2}|X_{u_1} = 0]$. We then assume that $\mathbf{E}[X_{u_i}|X_{u_1} = 1] > \mathbf{E}[X_{u_i}|X_{u_1} = 0]$ for $2 < i < n$. For the case of $i + 1$ we then have that for $k \in \{0, 1\}$

$$\begin{aligned}\mathbf{E}[X_{u_{i+1}}|X_{u_1} = k] =&\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 1]\mathbf{E}[X_{u_i}|X_{u_1} = k]+\\ &\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 0](1 - \mathbf{E}[X_{u_i}|X_{u_1} = k])\\ =&\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 0]+\\ &\mathbf{E}[X_{u_i}|X_{u_1} = k](\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 1]-\\ &\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 0]).\end{aligned}$$

Now we define $\delta = \mathbf{E}[X_{u_{i+1}}|X_{u_1} = 1] - \mathbf{E}[X_{u_{i+1}}|X_{u_1} = 0]$. Using the above result we have

$$\begin{aligned}\delta =&(\mathbf{E}[X_{u_{i+1}}|X_{u_i} = 1] - \mathbf{E}[X_{u_{i+1}}|X_{u_i} = 0])(\mathbf{E}[X_{u_i}|X_{u_1} = 1] - \mathbf{E}[X_{u_i}|X_{u_1} = 0])\\ &> 0.\end{aligned}$$

Above we used the induction hypothesis and the monotonicity of the follow probabilities in the overlap. □

3.1.3 Expected Follows on an Arborescence

We know that the optimal policy on a DAG is a linear extension. We look at the expected follows of the optimal policy on a subclass of DAGs known as arborescences. An arborescence is a DAG with a single root vertex, where all vertices have at most

one parent. In this case each vertex follows only one other vertex so the overlap of a vertex with the agent is either zero or one. Under this setting we can provide a closed form expression for the expected follows of the optimal policy.

We assume that the follow probability for a target vertex v with features set F_v and overlap ϕ_v is given by $p(\phi_v, F_v) = f(\phi_v)g(F_v)$. Let $\Delta = f(1) - f(0)$ and for each vertex v define its *susceptibility* as $g(F_v)$. Also, let $\mathcal{P}_l(v, G)$ be the set of cycle free directed paths of length l in a graph G which terminate on vertex v. For instance, if v is the root vertex of an arborescence G, then $\mathcal{P}_0(v, G) = \{v\}$, and if u is a child of the root, then $\mathcal{P}_0(u, G) = \{u\}$ and $\mathcal{P}_1(u, G) = \{(v, u)\}$. We have the following result for the expected follows of a linear extension policy (which is optimal) on an arborescence.

Theorem 3.1.3. *Let $G = (V, E)$ be an arborescence with N vertices. Let $J(G)$ be the expected follows of a linear extension policy on G. Then*

$$J(G) = f(0) \sum_{l=0}^{N-1} \Delta^l \sum_{v \in V} \sum_{p \in \mathcal{P}_l(v,G)} \prod_{u \in p} g(F_u). \tag{3.1}$$

This expression weighs each possible follow path of length l by Δ^l and the product of susceptibilities of vertices on the path. While this expression gives the exact number of expected follows on an arborescence where vertices have different susceptibilities, it can be greatly simplified if we assume all susceptibilities are uniform. For an arborescence, each vertex can be classified by its distance to the root. Let Z_i be the number of vertices a distance i from the root. Then we have the following result.

Lemma 3.1.4. *Let $G = (V, E)$ be an arborescence with N vertices. Assume that for all vertices $v \in V$, $g(F_v) = 1$ Let $J(G)$ be the expected follows of a linear extension policy on G. Then*

$$J(G) = \frac{f(0)N}{1 - \Delta} - \frac{f(0)\Delta}{1 - \Delta} \sum_{i=0}^{N-1} Z_i \Delta^i. \tag{3.2}$$

In the limiting case where $\Delta = 0$, $J(G) = f(0)N$. This means that when there is

no overlap boost, the expected follows is just the number of vertices multiplied by the constant follow probability $f(0)$. This formula also provides insights on what graph topologies are more conducive to follows. Consider two simple graphs. One is a path graph on N vertices, which we will refer to as P_N, and the other is a star graph with a single root with $N-1$ children, which we will refer to as S_N. For P_N, $Z_i = 1$ for $0 \leq i \leq N-1$ and for S_N $Z_0 = 1$ and $Z_1 = N-1$. Using equation (3.14) we find that

$$J(P_N) = \frac{f(0)N}{1-\Delta} - \frac{\Delta f(0)(1-\Delta^N)}{(1-\Delta)^2}$$
$$J(S_N) = \frac{f(0)N}{1-\Delta} - \frac{\Delta f(0)(1+\Delta(N-1))}{1-\Delta}$$

Taking the difference of these two expressions we find that

$$J(P_N) - J(S_N) = \frac{\Delta f(0)}{1-\Delta}\left(\frac{1-\Delta^N}{1-\Delta} - 1 - \Delta(N-1)\right).$$

Because $0 < \Delta < 1$, this difference is positive, so a path topology is better for getting follows than a star. This is expected because a path allows the agent multiple opportunities to gain overlap, whereas in the star, if the root does not follow, there will be no overlap for subsequent interactions. This analysis gives us the insight that in general, DAGs with longer paths produce more follows.

of Theorem 3.1.7. We make use of the following lemma.

Lemma 3.1.5. *Let $G = (V, E)$ be an arborescence rooted at u_1. Consider the path $(u_1, u_2, ..., u_i)$ in G and assume the agent interacts with the vertices in sequence along this path. Let X_i be the random variable which is one if u_i follows as a result of the agent interaction sequence and zero otherwise. Then*

$$\mathbf{E}[X_i] = f(0)\sum_{l=0}^{n-1}\Delta^l \sum_{p\in\mathcal{P}_l(u_i,G)}\prod_{u\in p} g(F_u). \tag{3.3}$$

With this lemma we can easily calculate the expected follows of a linear extension

policy on G. Note that because G is an arborescence, each vertex has a unique path from the root vertex. Therefore, we can directly apply Lemma 3.1.8 to each vertex using this path and then sum over all vertices. Doing so we obtain

$$\begin{aligned} J(G) &= \sum_{v \in V} \mathbf{E}\left[X_v\right] \\ &= f(0) \sum_{l=0}^{n-1} \Delta^l \sum_{v \in V} \sum_{p \in \mathcal{P}_l(v,G)} \prod_{u \in p} g(F_u). \end{aligned}$$

□

of Lemma 3.1.4. If we set the susceptibilities $g(F_u)$ to one for all vertices, the expected follows reduces to

$$J(G) = f(0) \sum_{l=0}^{N-1} \Delta^l \sum_{v \in V} |\mathcal{P}_l(v,G)|. \tag{3.4}$$

We need to count the number of vertices that are the final vertex in a directed path of length l for $0 \leq l \leq N-1$. This can be easily done by noting that the only vertices that end a path of length l must be at a distance of l or greater from the root. That is, there are $\sum_{i=l}^{N-1} Z_i$ vertices that terminate a path of length l. Using this, the expected follows becomes

$$\begin{aligned} J(G) &= f(0) \sum_{l=0}^{N-1} \Delta^l \sum_{i=l}^{N-1} Z_i \\ &= f(0) \sum_{i=0}^{N-1} Z_i \sum_{l=0}^{i} \Delta^l \\ &= f(0) \sum_{i=0}^{N-1} Z_i \frac{1-\Delta^{i+1}}{1-\Delta} \\ &= \frac{f(0)N}{1-\Delta} - \frac{f(0)\Delta}{1-\Delta} \sum_{i=0}^{N-1} Z_i \Delta^i. \end{aligned}$$

Above we have used the fact that $\sum_{i=0}^{N-1} Z_i = N$. □

of Lemma 3.1.8. We prove the result by induction. To simplify notation, let the

feature set of u_i simply be F_i. For $i = 1$, we have $\mathbf{E}[X_1] = f(0)g(F_1)$. The induction hypothesis is $\mathbf{E}[X_i] = f(0)\sum_{l=0}^{N-1}\Delta^l \sum_{p\in\mathcal{P}_l(u_i,G)}\prod_{u\in p} g(F_u)$. For $i+1$ we have

$$\begin{aligned}
\mathbf{E}[X_{i+1}] =& \mathbf{E}[X_{i+1}|X_i = 1]\,\mathbf{E}[X_i] \\
&+ \mathbf{E}[X_{i+1}|X_i = 0]\,(1 - \mathbf{E}[X_i]) \\
=& \mathbf{E}[X_i]\,(E[X_{i+1}|X_i = 1] - E[X_{i+1}|X_i = 0]) \\
&+ \mathbf{E}[X_{i+1}|X_i = 0] \\
=& \Delta g(F_{i+1}) f(0) \sum_{l=0}^{N-1} \Delta^l \prod_{u\in\mathcal{P}_l(u_i)} g(F_u) + f(0)g(F_{i+1}) \\
=& f(0) \sum_{l=0}^{N-1} \Delta^l \prod_{u\in\mathcal{P}_l(u_{i+1},G)} g(F_u).
\end{aligned}$$

□

3.1.4 Optimal Policy on a Graph with a Single Cycle

We now consider a graph with N vertices which has a single directed cycle on n vertices and assume that the vertex susceptibilities are all equal to one. The vertices in the cycle are $\{u_1, u_2, ..., u_n\}$ and the edges of the cycle are $\{(u_1, u_2), (u_2, u_3), ..., (u_{n-1}, u_n), (u_n, u_1)\}$. We also assume each cycle vertex u_i is followed by D_i vertices who follow no one else. We refer to this set of vertices as $\mathcal{F}_i$ for cycle vertex u_i. An illustration of this graph structure is shown in Figure 3-2. This graph is not a DAG so a linear extension cannot be calculated. However, there are n possible DAGs, each one determined by which cycle node is the root. The optimal policy will then be determined by one of these DAGs.

To determine the optimal DAG we use the following notation. Let π_i be the cyclic shift of the sequence $\{1, 2, ..., n\}$ that has i as the first element. Each DAG of the graph is associated with one of the π_i with u_i as the root, and therefore, the optimal policy will correspond to one of the π_i. Denote the index of the optimal policy by i^*. The following lemma characterizes the optimal policy.

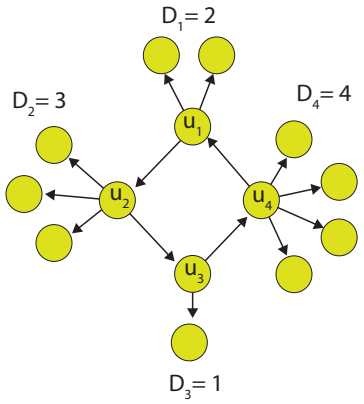

Figure 3-2: Example of a directed graph with a single cycle for $n = 4$.

Lemma 3.1.6. *Assume $f(F_v) = 1$ for $v \in V$. Then the index of the optimal policy for the follow-back problem with initial graph which contains a single n vertex cycle is*

$$i^* = \arg \min_{1 \le i \le n} \sum_{l=1}^{n} \Delta^l D_{\pi_i(l)}. \tag{3.5}$$

Recall that $\Delta < 1$. We see from this theorem that what the optimal policy does is select a DAG such that the vertices closer to the root have a lower number of followers. This makes intuitive sense because one would prefer to hold off on a cycle vertex with many followers because before attempting connect with these followers, one would like to maximize the overlap with the parent vertex. As an example, for the graph in Figure 3-2, for $0 < \delta < 1$ the optimal policy corresponds to the DAG rooted at vertex u_3 which also has the lowest number of followers among the cycle vertices.

of Lemma 3.1.6. Let G_i be the DAG rooted at u_i. This DAG is also an arborescence, so we can use equation (3.14) to calculate the expected follows. The sequence of degrees of the cyclic vertices in G_i is given by $\left\{D_{\pi_i(1)}, D_{\pi_i(2)}, ..., D_{\pi_i(n)}\right\}$. This degree sequence is what will determine the optimal DAG. For G_i, we have that $Z_0 = 1$, $Z_{n+1} = D_{\pi_i(n)}$, and $Z_l = 1 + D_{\pi_i(l)}$ for $1 \le l \le n$.

Applying equation (3.14) we get

$$J(G_i) = \frac{f(0)N}{1-\Delta} - \frac{f(0)\Delta}{1-\Delta}\left(N + \sum_{l=1}^{n} \Delta^l D_{\pi_i(l)}\right)$$
$$= f(0)N - \frac{f(0)\Delta}{1-\Delta}\sum_{l=1}^{n} \Delta^l D_{\pi_i(l)}.$$

To minimize this expression one selects the index i to minimize $\sum_{l=1}^{n} \Delta^l D_{\pi_i(l)}$. □

3.1.5 Expected Follows on a DAG

We know that the optimal policy on a DAG is a linear extension. A natural question to ask is what is the expected number of follows the optimal policy achieves on a DAG? Here we provide a closed form expression for the expected follows of the optimal policy, under an additional assumption on the overlap function $f(\phi_v)$. We assume that the follow probability for a target vertex v with features set F_v and overlap ϕ_v is given by $p(\phi_v, F_v) = f(\phi_v)g(F_v)$. To obtain a simple closed form expression, we assume that $f(\phi_v)$ is monotonically increasing and linear, i.e. $f(\phi_v) = \alpha + \beta\phi_v$ for some $\alpha, \beta > 0$. For each vertex v, we define its *susceptibility* as $g(F_v)$. Also, let $\mathcal{P}_l(v, G)$ be the set of cycle free directed paths of length l in a graph G which terminate on vertex v. For instance, if v is a root vertex of a DAG G, then $\mathcal{P}_0(v, G) = \{v\}$, and if u is a child of v, then $\mathcal{P}_0(u, G) = \{u\}$ and $\mathcal{P}_1(u, G) = \{(v, u)\}$. We have the following result for the expected follows of a linear extension policy (which is optimal) on a DAG.

Theorem 3.1.7. *Let $G = (V, E)$ be a DAG with N vertices. Let the follow probability for a target vertex v with features set F_v and overlap ϕ_v be given by $p(\phi_v, F_v) = f(\phi_v)g(F_v)$ and let $f(\phi_v) = \alpha + \beta\phi_v$ for some $\alpha, \beta > 0$. Let $J(G)$ be the expected follows of a linear extension policy on G. Then*

$$J(G) = \alpha \sum_{l=0}^{N-1} \beta^l \sum_{v \in V} \sum_{T \in \mathcal{P}_l(v,G)} w_T \tag{3.6}$$

where for a path T we define

$$w_T = \prod_{u \in T} g(F_u). \tag{3.7}$$

3.1.6 Proof of Theorem 3.1.7

We make use of the following lemma.

Lemma 3.1.8. *Let $G = (V, E)$ be a DAG with N vertices. Let the follow probability for a target vertex v with features set F_v and overlap ϕ_v be given by $p(\phi_v, F_v) = f(\phi_v)g(F_v)$ and let $f(\phi_v) = \alpha + \beta\phi_v$ for some $\alpha, \beta > 0$. Assume the agent interacts with the parents of a vertex v before it interacts with v. Let X_v be the random variable which is one if v follows as a result of the agent interaction sequence and zero otherwise. Also, let $P(v)$ denote the set of parent vertices of v. Then*

$$\mathbf{E}\left[X_v\right] = \alpha \sum_{l=0}^{N-1} \beta^l \sum_{p \in \mathcal{P}_l(u_i, G)} \prod_{u \in p} g(F_u). \tag{3.8}$$

With this lemma we can easily calculate the expected follows of a linear extension policy on G. By directly applying Lemma 3.1.8 to each vertex and then summing over all vertices. Doing so we obtain

$$\begin{aligned} J(G) &= \sum_{v \in V} \mathbf{E}\left[X_v\right] \\ &= \alpha \sum_{l=0}^{n-1} \beta^l \sum_{v \in V} \sum_{p \in \mathcal{P}_l(v, G)} \prod_{u \in p} g(F_u). \end{aligned}$$

3.1.7 Proof of Lemma 3.1.8

We require the following result.

Lemma 3.1.9. *Consider a vertex v in a DAG $G = (V, E)$ with parents given by the set $P(v) \subset V$, $|P(v)| = n$. Let the susceptibility and overlap functions be g and f, respectively. Assume the vertices are interacted with according to a linear extension*

policy π. *Let* $q_v = \mathbf{E}[X_v|\pi]$. *Then we have that*

$$q_v = g(F_v) \sum_{k=0}^{n} \Delta_k \sum_{S \subseteq P(v): |S|=k} \prod_{u \in S} q_u \tag{3.9}$$

where we define $\Delta_k = \sum_{i=0}^{k} a_i^k f(i)$ *where* $a_i^k = a_{i-1}^{k-1} - a_i^{k-1}$, $a_{k+1}^k = 0$, $a_k^k = 1$, $a_0^k = (-1)^k$, *and* $a_{-1}^k = 0$.

If we assume that $f(\phi) = \alpha + \beta\phi$, then equation (3.9) takes a simpler form. We use the following result regarding the terms a_i^k.

Lemma 3.1.10. *Let the terms* a_i^k *be given in Lemma 3.1.9. Then we have that* $\sum_{i=0}^{k} a_i^k = 0$ *for* $k \geq 0$, $\sum_{i=0}^{1} a_i^1 i = 1$, *and* $\sum_{i=0}^{k} a_i^k i = 0$ *for* $k > 1$.

For linear f, the terms Δ_k take the following form.

$$\Delta_k = \alpha \sum_{i=0}^{k} a_i^k + \beta \sum_{i=0}^{k} a_i^k i$$

Using Lemma 3.1.10 we find that

$$\begin{aligned} \Delta_0 &= \alpha \\ \Delta_1 &= \beta \\ \Delta_k &= 0, \quad k > 1. \end{aligned}$$

Substituting this into equation (3.9) we obtain

$$q_v = g(F_v) \left(\alpha + \beta \sum_{u \in P(v)} q_u \right) \tag{3.10}$$

This expression is a recursion in the depth of the DAG. If we expand it we obtain

$$\begin{aligned}
q_v =& g(F_v)\left(\alpha + \beta \sum_{u\in P(v)} g(F_u)\left(\alpha + \beta \sum_{w\in P(u)} q_w\right)\right) \\
=& \alpha g(F_v) + \alpha\beta \sum_{u\in P(v)} g(F_u)g(F_v) + \beta^2 \sum_{u\in P(v)}\sum_{w\in P(u)} q_w g(F_u)g(F_v) \\
=& \alpha g(F_v) + \alpha\beta \sum_{u\in P(v)} g(F_u)g(F_v) + \beta^2 \sum_{u\in P(v)}\sum_{t\in P(u)} g(F_t)g(F_u)g(F_v)\left(\alpha + \beta \sum_{s\in P(t)} q_s\right) \\
=& \alpha g(F_v) + \alpha\beta \sum_{u\in P(v)} g(F_u)g(F_v) + \alpha\beta^2 \sum_{u\in P(v)}\sum_{t\in P(u)} g(F_t)g(F_u)g(F_v) \\
&+ \beta^3 \sum_{s\in P(t)} q_s g(F_t)g(F_u)g(F_v) \\
=& \alpha\left(\sum_{(u)\in\mathcal{P}_0(v,G)} g(F_u) + \sum_{(u,v)\in\mathcal{P}_1(v,G)} g(F_u)g(F_v) + \sum_{(t,u,v)\in\mathcal{P}_3(v,G)} g(F_t)g(F_u)g(F_v)\right) \\
&+ \beta^3 \sum_{(s,t,u,v)\in\mathcal{P}_4(v,G)} q_s g(F_t)g(F_u)g(F_v).
\end{aligned}$$

If we continue this expansion higher up the DAG, then this pattern continues until we hit the roots, at which point there are no more parents and the expansion terminates, since for a root r, we have trivially that $q_r = \alpha g(F_r)$. What is happening is then evident. The expression of q_v is summing over all paths in the DAG which terminate on v. Each such path p containing l vertices is weighted by $\alpha\beta^{l-1}\prod_{u\in p} g(F_u)$. Using this, we obtain

$$q_v = \alpha \sum_{l=0}^{N} \beta^l \sum_{p\in\mathcal{P}_l(v,G)} \prod_{u\in p} g(F_u). \tag{3.11}$$

The total follows on the DAG is given by

$$\begin{aligned}
J(G) &= \sum_{v\in V} q_v \\
&= \alpha \sum_{l=0}^{N}\sum_{v\in V} \beta^l \sum_{p\in\mathcal{P}_l(v,G)} \prod_{u\in p} g(F_u).
\end{aligned}$$

3.1.8 Proof of Lemma 3.1.9

We prove the result by induction. The induction hypothesis is given by equation (3.9). We assume that v has n parents. For $n = 0$, we have $q_v = f(0)g(F_v)$, which satisfies the induction hypothesis. For $n = 1$, let the parent be u. From the definition in Lemma 3.1.9 we have that $\Delta_0 = f(0)$ and $\Delta_1 = f(1) - f(0)$. Then we have

$$\begin{aligned} q_v &= g(F_v)\left(f(0)(1-q_u) + f(1)q_u\right) \\ &= g(F_v)\left(f(0) + q_u(f(1) - f(0))\right) \\ &= g(F_v)\left(\Delta_0 + \Delta_1\right) \end{aligned}$$

which also satisfies the induction hypothesis. For the $n+1$ case, let the additional vertex added be denoted $n+1$. Conditional on this vertex not following, q_v is not changed. Conditional on this vertex following, the overlap increases by one. The value of q_v, conditioned on this event, is given by equation (3.9), except that the argument of f is increased by one in the definition of Δ_k. To ease our notation, let us define $\Delta'_k = \sum_{i=0}^{k} a_i^k f(i+1)$. We let $P(v)$ be the parents of v not including $n+1$. Then we have

$$\begin{aligned} q_v =& \mathbf{E}[X_v | X_{n+1} = 0](1 - q_{n+1}) + \mathbf{E}[X_v | X_{n+1} = 1] q_{n+1} \\ =& (1 - q_{n+1}) g(F_v) \sum_{k=0}^{n} \Delta_k \sum_{S \subseteq P(v): |S| = k} \prod_{u \in S} q_u \\ &+ q_{n+1} g(F_v) \sum_{k=0}^{n} \Delta'_k \sum_{S \subseteq P(v): |S| = k} \prod_{u \in S} q_u \\ =& g(F_v) \sum_{k=0}^{n} \Delta_k \sum_{S \subseteq P(v): |S| = k} \prod_{u \in S} q_u \\ &+ q_{n+1} g(F_v) \sum_{k=0}^{n} (\Delta'_k - \Delta_k) \sum_{S \subseteq P(v): |S| = k} \prod_{u \in S} q_u \end{aligned} \tag{3.12}$$

The difference $\Delta'_k - \Delta_k$ is given by

$$\begin{aligned}
\Delta'_k - \Delta_k &= \sum_{i=0}^{k} a_i^k (f(i+1) - f(i)) \\
&= a_k^k f(k+1) - a_k^k f(k) + a_{k-1}^k f(k) - a^k + k - 1 f(k-1) + ... + a_0^k f(1) - a_0^k f(0) \\
&= f(k+1)(a_k^k - a_{k+1}^k) + f(k)(a_{k-1}^k - a_k^k) + ... + f(0)(a_0^k - a_{-1}^k) \\
&= \sum_{i=0}^{k+1} a_i^{k+1} f(i) \\
&= \Delta_{k+1}.
\end{aligned}$$

Above we used the definition of a_i^k given in Lemma 3.1.9. Substituting this into equation (3.12) and letting $P'(v)$ denote the parents of v including $n+1$, we obtain

$$\begin{aligned}
q_v =& g(F_v) \sum_{k=0}^{n} \Delta_k \sum_{S \subseteq P(v): |S|=k} \prod_{u \in S} q_u + q_{n+1} g(F_v) \sum_{k=0}^{n} \Delta_{k+1} \sum_{S \subseteq P(v): |S|=k} \prod_{u \in S} q_u \\
=& g(F_v) \sum_{k=0}^{n} \Delta_k \sum_{S \subseteq P(v): |S|=k} \prod_{u \in S} q_u + \sum_{k=1}^{n+1} \Delta_{k+1} \sum_{S \subseteq P'(v): |S|=k} \prod_{u \in S} q_u \\
=& g(F_v) \sum_{k=0}^{n+1} \Delta_k \sum_{S \subseteq P'(v): |S|=k} \prod_{u \in S} q_u.
\end{aligned}$$

The above expression matches the induction hypothesis, completing the proof.

This expression weighs each possible follow path T of length l by $\beta^l w_T$. The expected number of follows is obtained by summing over all possible paths in the DAG. If we look at each term corresponding to a path length l in the expression for $J(G)$ we can gain a better understanding of how the overlap plays a roll in the expected follows. For each value l we define

$$J^l(G) = \alpha \beta^l \sum_{v \in V} \sum_{T \in \mathcal{P}_l(v,G)} w_T \tag{3.13}$$

as the contribution to the total expected follows $J(G)$ coming from a specific value of

l in the sum in equation (3.6). With this notation, we can rewrite equation (3.6) as

$$J(G) = \sum_{l=0}^{N-1} J^l(G). \tag{3.14}$$

For $l = 0$ we have

$$J^0(G) = \alpha \sum_{v \in V} g(F_v).$$

That is, the zeroth order contribution to $J(G)$ is the sum of the vertex susceptibilities. This makes sense, as a vertex is a one hop path. The value $J^0(G)$ is the expected follows that would be obtained if the overlap had no impact of the follow probability. The first order term is

$$J^1(G) = \alpha\beta \sum_{(u,v) \in E} g(F_u)g(F_v),$$

which is simply the sum over all the edges in the graph, with edge weights given by the product of the susceptibility of their end vertices. Edges can also be thought of as the one-hop paths in the graph. Continuing this way we obtain contributions from higher order paths, up to paths of length $N - 1$ (the longest path possible in a graph with N vertices). A useful computational question is how many terms to keep in this sum when actually calculating $J(G)$. This is entirely dependent upon the density of the graph and the value of the susceptibilities and overlap function. To understand this question, let us assume that all susceptibilities are equal to some value γ. Let us also define N_l as the number of paths in a graph of length l. With this notation, N_0 is the number of vertices and N_1 is the number of edges. For ease of notation, we define $N = N_0$. Under this scenario, all paths of equal length have the same weight. Therefore, we can write $J^l(G)$ as

$$J^l(G) = \alpha\gamma(\beta\gamma)^l N_l.$$

The first term $\alpha\gamma$ is the same for all l and can be ignored. The term $\beta\gamma < 1$ because the follow probability must be less than or equal to one and we assume $\alpha > 0$. The term N_l depends upon the density of the graph. For a sparse graph N_l will scale like

N, but for a dense graph we can have N_l scaling like N^l. Therefore, for sparse graphs $J^l(G)$ will decay exponentially fast in $\beta\gamma$, while for denser graphs it will scale like $(\beta\gamma N)^l$ which can decay or increase depending upon the value of $\beta\gamma$ relative to n. If the overlap is weak (small β) or the graph is small, then $\beta\gamma n$ will be less than one and the higher order terms will be much smaller than the lower order terms. If the overlap is strong (large β) or the graph is large, then higher order terms may not be negligible.

Chapter 4

Graph Heuristic

4.1 Optimal Policy on a Directed Graph

The linear extension policy for a DAG does not apply to a general directed graph because there is more than one DAG for the graph due to the presence of cycles. However, the DAG analysis suggests that whatever the optimal policy is, it must not violate the order relations imposed by the graph. In fact, a simple application of Lemma 3.1.2 gives us the following result, which we state here without proof.

Lemma 4.1.1. *Let π^* be the optimal policy for an arbitrary directed graph $G = (V, E)$ and let $u, v \in V$. If there is a directed path in G from u to v, but not from v to u, then u must come before v in π^*.*

This lemma is a generalization of Theorem 3.1.1 to arbitrary graphs. It says that if there is a path between two vertices in one direction, but not in the reverse, then the vertex at the start of the path should always be followed before the other. Furthermore, because the optimal policy must respect the order imposed by the graph, then the optimal policy should be a linear extension of a DAG of the underlying graph. Because all linear extensions of a DAG are optimal, finding the optimal policy reduces to finding the optimal DAG. Formally, let $\mathcal{D}(G)$ be the set of all possible DAGs of G. The optimal DAG is the one that maximizes the expected follows. That is, the

optimal DAG is the solution to the following optimization problem.

$$\max_{D \in \mathcal{D}(G)} J(D). \tag{4.1}$$

There are some features the optimal DAG should have. Consider the expression for $J(G)$ for a DAG G from equation (3.14). The term $J^0(G)$ is independent of the DAG chosen as long as all vertices are included. However, $J^1(G)$ will be larger if there are more edges with heavier weights (product of end vertex susceptibilities). The same goes for larger values of l. This suggests that when searching for the optimal DAG, one would like to have as many "heavy" edges as possible. In the case when the susceptibilities are uniform, this reduces to having a DAG with as many edges as possible. This is known as the minimum feedback arcset problem and is well known to be NP-hard to solve [23]. Therefore, finding an exact solution to (4.1) may not be easily done. However, an exactly optimal solution may not be necessary, as the higher order terms in $J(G)$ may not be significant, depending upon the graph structure and follow probabilities. Instead, heuristic solutions may be sufficient for practice. We will look at such heuristic policies next.

Our analysis thus far has shown that the optimal policy on a graph should correspond to a DAG of the underlying graph. We have also seen how to calculate expected follows on a DAG from Theorem 3.1.7. In addition, by studying the structure of the expression for the expected follows we have seen that more edges in the DAG can lead to increased expected follows. Combining these results we now propose a heuristic policy for an arbitrary directed graph G. Give each vertex $v \in V$ a weight g_v. Weight the edges of G by the product of its incident vertices. So, an edge (i, j) has weight $g_i g_j$. Then form a DAG from G as follows. Let B be the maximum branching forest of G. Recall that a maximum branching forest for a directed graph is a subgraph such that each vertex has at most one parent and the edges included in the subgraph are of maximal weight. Then add edges to B with the following heuristic.

Create a list of edges E of G such that they are not in the maximum branching. Then while this list is non-empty, do the following. Select the vertex v of of B with

smallest in-degree, ties broken arbitrarily. Then select the edge from E incident to v with maximal weight. Update B by adding this edge to the graph. If the updated graph contains a cycle, we remove the newly added edge. Remove the edge from E and continue. Each edge added has a chance of creating a cycle. We would like to add as many edges as possible to the graph. By attempting to append the edges in a smart mannner, we minimize the chance that each new edge forms a cycle.

The study of Directed Acyclic Graphs (DAGs) overlaps with the study of partially ordered sets, since a partial order may be imposed on the nodes of a DAG. Constraining the graph the graph to be acyclic ensures that it still constitutes a valid partial order. [33] discusses the process of "refining" a partial order. By appending edges from the parent "friends" graph G onto B, the heuristic is refining the partial order imposed on the nodes of the graph, making it more precise. We are also getting a better estimate of the true expected follows of a linear extension policy.

Let us refer to DAG produced as a result of this procedure as D_v. We will define the *follow-back score* of D_v to be the expected follows on the DAG, i.e. $J(D_v)$, as given by equation (3.6). The heuristic attempts to incorporate edges in a smart way to increase the expected follows of the DAG. Then our follow-back policy will be to interact with the vertices of G in a sequence that is a linear extension of this DAG. While this heuristic is not guaranteed to produce the optimal DAG, we will see in Section 4.2 that the policies we obtain perform well.

4.2 Results

In this section we compare our follow-back score heuristic policy to several other policies on real Twitter graphs. The policies we compare with are based on a various network centrality measures. We use simulations to evaluate the policies with follow probabilities given by the results of our empirical analysis.

ALGORITHM 1: Follow-back

Input: Target friend graph $G = (V, E)$;
Maximal branching $B = (V, E_B)$ of G;
while $E \neq \emptyset$ **do**
    Order E;
    $e = E[0]$;
    if $e \notin E_B$ **then**
        $E_B = \{e\} \cup E_B$;
        if B *is cyclic* **then**
            $E_B = E_B \setminus \{e\}$
        else
            continue
        end
    else
        continue;
    end
end
Output: DAG B.

4.2.1 Simulation Methodology

We evaluate the performance of a policy on a Twitter graph as follows. We simulate an interaction with the the first vertex in the policy. The result of this interaction is given by a Bernoulli random variable that is one with probability given by the follow probability from 3.1.1. We then update the overlap of any other vertex in the graph as a result of this interaction. In particular, if this vertex follows back, then all of its children in the graph increase their overlap by one. We note that all edges of the graph are used to simulate the follows. We repeat this process for each vertex in the policy until all vertices have been interacted with. When we reach the end of the policy, we evaluate how many follows were received. We repeat this process for $10,000$ trials to obtain the distribution of the follows of the policy. For our follow probability model, we use the coefficients learned in our regression 2.4. However, we decided to remove the coefficient B_0 from the model in simulation. The reason we did this was to inflate the prior follow probabilities of each target. Since the agents that we used to learn the follow probability model used the Twitter programming interface to automatically post content, we assumed that their ability to gain followers would be

Table 4.1: Twitter User Descriptions

User	Description
@billmcraven	Chancellor of University of Texas System
@DJ44	Chief Data Sceintist of the United States Office of Science and Technology Policy
@McKinsey	Management consulting firm
@MIT	Institution of higher education
@ORCenter	Academic department within institution of higher education
@POTUS	United States President
@realDonaldTrump	Businessman and presidential candidate
@taylorswift13	Celebrity singer
@zlisto	Business school professor

significantly less than a human operator who could control the account and interact with greater skill.

4.2.2 Twitter networks

Our test graphs for the simulation analysis are the "friends graph" of nine Twitter users, ranging from different types of accounts and levels of public notoriety. We list brief descriptions of each user in 4.1. Their friends graphs have very different structural properties which we list in 4.2. Friends graphs for celebrity users such as @taylorswift13 tend to be more dense. This is because celebrities tend to follow other celebrities who, in turn, follow one another. Also, the median follower count of the vertices in celebrity friend graphs like @taylorswift13's graph is two orders of magnitude larger than typical users like the MIT academic department's account, @ORCenter. This is because most people @taylorswift13 follows are also celebrities with large Twitter followings. The result of this is that we expect the susceptibilities to be much lower for the vertices of in celebrities' graphs than in typical users' graphs. This means that @taylorswift13's graph will be harder to penetrate and we expect to get fewer follows compared with a graph that of @ORCenter.

4.2.3 Network centrality policies

We compare our follow-back score policy to several other network centrality based policies. Network centralities are functions that map a vertex in a graph to a real

Table 4.2: Twitter Network Topologies

Friends graph	@taylorswift13	@ORCenter	@billmcraven	@DJ44	@POTUS	@realDonaldTrump	@zlisto	@McKinsey	@MIT
Number of vertices	245	127	30	280	71	42	319	318	320
Number of edges	5589	792	323	5502	2195	523	537	3519	4182
Density	0.09	0.05	0.37	0.07	0.44	0.30	0.005	0.03	0.04
Mean degree	22.81	6.24	10.77	19.65	30.92	12.45	1.68	11.07	13.07
Max out degree	115	41	27	104	61	34	25	62	58
Min out degree	0	0	1	0	0	1	0	0	0
Max in degree	149	66	28	76	62	35	42	80	152
Min in degree	0	0	1	0	0	0	0	0	0
Number weak components	2	22	1	3	1	1	135	6	8
Largest component diameter	6	5	3	8	3	3	3	10	7
Sum of vertex susceptibilities	23.95	50.67	7.45	66.50	8.00	5.49	85.34	68.37	74.42

number. There are several different network centralities to choose from, such as degree centrality, eigenvector centrality, or betweenness centrality. For each network centrality, we use the following policy: rank the vertices in order of increasing or decreasing network centrality. The intuition for this policy comes from the fact that network centralities give scores to vertices that reflect their importance in different graph metrics. Our network centrality policies simply interact with the vertices in either increasing or decreasing importance with respect to the chosen network centrality.

In addition to the network centrality based policies, we also look at a random policy, where the policy is a random permutation of the vertices. This policy serves as a baseline and represents how many follows would be achieved without using any information about the graph structure or vertex susceptibility.

4.2.4 Simulation Results

We test the different policies (network centralities, follow-back algorithm, and random) on the friends graph of the users listed in 4.1. The results are shown in Table 4.3, where each number is the average number of follows gained in the user's friend's graph over 10,000 simulations. As can be seen, the follow-back score policy is frequently one of the top performing policies, and there is a very small difference between the expected follows of the top policies. Also, "smart " policies achieve as much as 10% more follows than their inverse order. This shows the scale of gain possible using intelligent interaction policies. We note that the gain over random is roughly the same for most graphs. This is not surprising if one considers the form of the expression for the expected follows. The random policy DAG most likely does not contain many edges. The expected follows of this policy would be close to $J^0(G)$. The follow-back score policy creates a DAG with more edges, so to first order their value can be expressed as $J^0(G) + J^1(G)$. The fractional gain of the follow-back score policy over random is the approximately given by $J^1(G)/J^0(G)$. Expanding the ratio

of these quantities we find

$$\frac{J^1(G)}{J^0(G)} = \beta \frac{\sum_{(u,v)\in E} g(F_u)g(F_v)}{\sum_{v\in V} g(F_v)} \tag{4.2}$$

We see that this ratio is proportional to β. In our simulations, the value of β is 0.29. If we assume that vertex susceptibilities all equal γ, then the ratio simplifies to

$$\frac{J^1(G)}{J^0(G)} = \beta\gamma \frac{|E|}{|V|}. \tag{4.3}$$

For very dense graphs, there should be more gain compared to sparser graphs. Also, the gain should be higher for graphs where the vertices have a higher susceptibility. This is what we observe in Table 4.3. The gains are roughly the same for both graphs because while the potential for gain in denser graphs is higher, denser graphs tend to be associated with "celebrity" users who have lower susceptibilities. Sparser graphs, on the other hand, are more commonly associated with typical users who tend to have higher susceptibilities. Therefore, these effects combine to give similar gains for a majority of the graphs. One reason why many of the network centrality policies perform well is because their policies are highly correlated.

Table 4.3: Expected follows of different policies on the friends graphs.

Policy	@taylorswift13	@ORCenter	@billmcraven	@DJ44	@McKinsey	@MIT	@POTUS	@realDonaldTrump	@zlisto
Follow-back	34.49	66.18*	10.49	136.37	90.57	93.14	20.87*	7.61	89.65*
Betweenness centrality (ascending)	35.96	61.67	10.00	108.15	85.06	110.00	18.57	7.80	88.42
Betweenness centrality (descending)	32.95	56.49	10.35	108.83	85.86	88.74	15.86	7.11	87.20
Closeness centrality (ascending)	33.55	54.88	9.89	102.81	82.14	93.32	18.16	7.39	86.39
Closeness centrality (descending)	36.63	63.69	10.54	119.62	90.59*	110.02	17.59	7.53	89.09
Degree centrality (ascending)	35.74	58.23	10.02	104.82	84.46	104.47	18.64	7.77*	88.58
Degree centrality (descending)	33.45	59.42	10.19	112.93	86.56	93.10	15.65	7.19	86.86
Eigenvector centrality (ascending)	38.00*	63.27	10.78*	113.44	88.29	111.54*	18.84	7.91*	88.96
Eigenvector centrality (descending)	31.39	54.38	9.64	112.13	82.96	87.04	15.48	6.94	86.41
In-degree centrality (ascending)	37.97	63.20	10.53	111.16	88.73	110.97	18.88	7.88	89.23
In-degree centrality (descending)	31.06	54.43	9.85	105.90	82.03	86.70	15.53	6.93	86.23
Load centrality (ascending)	35.75	61.78	9.87	108.02	84.84	109.85	18.51	7.80	88.56
Load centrality (descending)	33.12	56.28	10.37	108.75	85.98	88.71	15.93	7.10	87.02
Out-degree centrality (ascending)	33.51	55.08	9.89	99.93	81.17	92.82	17.83	7.45	86.65
Out-degree centrality (descending)	36.33	63.51	10.42	119.80*	90.49	110.22	17.63	7.54	88.89
Random	34.95	59.52	10.23	108.97	86.41	100.59	17.49	7.51	87.74

Chapter 5

General Followback

5.1 Set up and Motivation

We now look at a more general version of the follow-back problem. Given an arbitrary directed graph, an agent might want to gain follows from some subset of targets within the graph. This situation is likely in marketing or advertisement campaigns. Presumably, there are already individual users or other companies within the Twitter network that are deemed authority figures on the company's product or area of focus. These prominent users probably have large followings that the company undergoing the advertising campaign wishes to reach. If the company can gain followers among the "authority-like" users, it will have access to these user's pre-existing follower-bases. This is precisely the problem that we address in this chapter: when the agent seeks to gain follows from multiple targets simultaneously and is permitted to interact with the friends of the targets.

5.2 Integer Program

We use constrained integer programming to formulate the problem. We add a constraint that limits the number of interactions that the target is allowed and makes the problem more practically applicable. Despite the increased complexity of the problem, we use the intuition gained from earlier analysis to formulate our objective.

We seek to find a directed acyclic subgraph with the maximum number of paths of maximally weighted vertices that lead into the targets, subject to the number of vertices in the subgraph is less than the maximum number of interactions allowed. In the basic problem where all of the vertices in the graph are friends of the target, we found that deriving an optimal policy involved selecting a directed acyclic subgraph of maximal followback score. Recall that followback score was calculated by taking a sum of all the vertex susceptibility weighted dot products of the paths in the graph as shown in 3.6. The objective function in our integer program is doing the same thing; however, since a general graph might include vertices that are not friends of the targets, we want to restrict our focus to paths that specifically end on one of the targets. Since the acyclic subgraph constitutes a valid partial order, we can, as before, derive an optimally sequenced interaction policy by taking a linear extension of the graph.

For an arbitrary directed graph G of N vertices, give each vertex $v \in V$ a weight g_v. Then form a DAG G^* from G using the integer program, relying upon notation established in 3.1.5. The objective function corresponds to 3.6 and the constraints limit the maximum number of interactions that the agent is allowed to c, ensure that the graph is acyclic, and guarantee that selected edges have incident vertices that are also selected. This DAG is optimal for the problem, and 4.1 showed us that any linear extension of the DAG is an optimal interaction sequence.

$$x_i = \begin{cases} 1, & \text{if } v_i \in G^* \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

$$y_{i,j} = \begin{cases} 1, & \text{if } e_{i,j} \in G^* \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

$$\max_{\mathbf{x},\mathbf{y}} \sum_{i \in V} g_i x_i + \sum_{t \in R} \sum_{l=1}^{N-1} \sum_{T \in \mathcal{P}_l(t,G)} \beta^l \prod_{u \in T} g_u \prod_{(i,j) \in T} y_{ij}$$

Subject to

$$\begin{aligned}
&\sum_{i \in V} x_i \leq c \\
&y_{i,j} \leq E_{i,j} && \forall (i,j) \in E \\
&\sum_{(i,j) \in \mathcal{C}} y_{i,j} \leq |\mathcal{C}| - 1 && \forall \mathcal{C} \\
&x_i - y_{i,j} \geq 0 && \forall i,j \\
&x_j - y_{i,j} \geq 0 && \forall i,j \\
&x_i \in \{0,1\} \\
&y_{i,j} \in \{0,1\}
\end{aligned} \tag{5.3}$$

5.3 Tractability Issues

Our integer program is highly nonlinear. The nonlinear terms come about from the inclusion of higher order paths in the objective function. The authors are not aware of any solvers that would be able to handle such a problem in reasonable time. We will not analyze the complexity of the general integer program, but rather, choose to focus on viable integer-programming based heuristics that exhibit good performance.

Recall that the objective function 5.3 sums over all of the paths in the DAG and weights them by the vertex susceptibilities along the path as well as a constant term that decreases exponentially in path length. We consider all weighted paths of length l included in the objective function to be lth order contribution to the objective value. Note that 5.3 includes paths of all lengths, which, for an N node graph, could be as large as the $N-1th$ order. The higher order terms in the function give rise to the high nonlinearity of the function since they must include products of the binary edge variables $y_{ij} \forall (i,j)$ in each path.

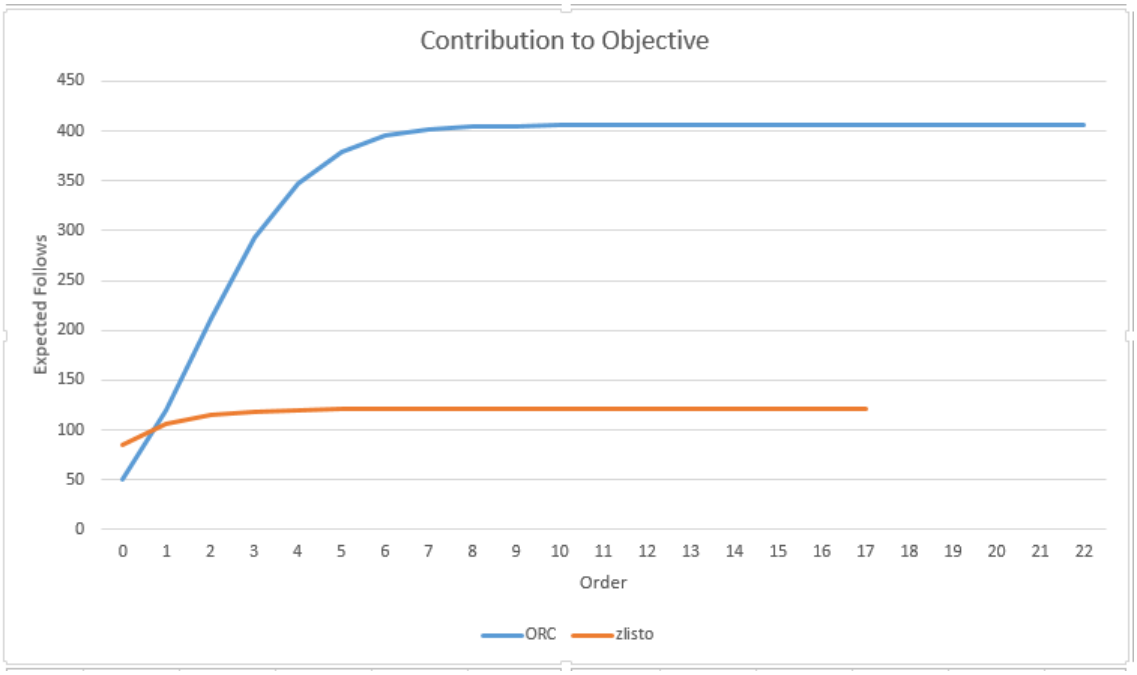

Figure 5-1: Contribution to objective value by order for two accounts.

We explore the contribution to the objective value that each order term provides for two Twitter graphs that were small enough to allow us to calculate the expected follows completely on their induced DAGs in 5-1. Approximately 90% of the objective value for the zlisto account is captured by the zero and first order contributions of the objective value and more than 99% of the objective value is captured by the second order. On the other hand, the first order contribution only captures approximately 25% of the objective value for the ORC account and 99% of the value is not captured until the seventh order. We did not find these results surprising since the ORC friends graph is 100x more dense than the zlisto friends graph. In general, we expect more dense graphs to have a greater proportion of their objective value captured by higher order paths, since a greater number of longer paths exist in these graphs.

5.4 Integer Program based heuristic

5.4.1 Minimum Feedback Arc Set

Despite the impact that higher order terms have in the objective value on dense graphs, we decide to truncate the objective function after the second order terms. We argue that optimizing over lower order terms exclusively is acceptable firstly because of tractability. Solving our objective to optimality in the second order is called the minimum feedback arcset problem and is well known to be NP-hard to solve [23]. There is an abundance of research on the problem and methods to solve it using integer programming exist [2]. We also legitimize focusing on lower order terms in our optimization because we saw that for low density graphs, lower order terms carry a high proportion of the objective value 5.3, and many of the social media networks of interest will be of low density.

5.4.2 Formulation

We use graph centrality measures as a surrogate to capture some of the granularity in objective value lost by ignoring these terms. Given a graph centrality score c_i for vertex i, we reformulate the integer program as follows. Note that we decrease the nonlinearity in the objective by optimizing over one and two-hop-paths alone.

$$x_i = \begin{cases} 1, & \text{if } v_i \in G^* \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

$$y_{i,j} = \begin{cases} 1, & \text{if } e_{i,j} \in G^* \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

$$\max_{\mathbf{x},\mathbf{y}} \sum_{i \in V} g_i x_i + c_i x_i + \sum_{t \in R} \sum_{l=1}^{2} \sum_{T \in \mathcal{P}_l(t,G)} \beta^l \prod_{u \in T} g_u \prod_{(i,j) \in T} y_{ij}$$

Subject to

$$\begin{aligned}
&\sum_{i \in V} x_i \leq c && \\
&y_{i,j} \leq E_{i,j} && \forall (i,j) \in E \\
&\sum_{(i,j) \in \mathcal{C}} y_{i,j} \leq |\mathcal{C}| - 1 && \forall \mathcal{C} \\
&x_i - y_{i,j} \geq 0 && \forall i,j \\
&x_j - y_{i,j} \geq 0 && \forall i,j \\
&x_i \in \{0,1\} && \\
&y_{i,j} \in \{0,1\} &&
\end{aligned} \tag{5.6}$$

The matrix associated with the second order portion of the objective is not generally positive semi-definite. To ensure that optimization solvers can handle the formulation, we seek to make the objective linear. We add additonal binary variables $z_{i,j,k}$ to be associated with two hop paths (i,j,k). This avoids the nonlinearitry in 5.6 when we multiply $y_{i,j}y_{j,k}$ to represent the this path. At the same time, it increases the number of binary variables in our formulation exponentially. For an N vertex graph, this would yield N^3 binary variables.

$$\max_{\mathbf{x},\mathbf{y},\mathbf{z}} \sum_{t \in R} \sum_{i \in V} g_i x_i + c_i x_i + \beta g_i g_t y_{i,t} + \beta^2 \sum_{j \in V} g_i g_j g_t z_{i,j,t}$$

Subject to

$$\begin{aligned}
&\sum_{i \in V} x_i \leq c \\
&y_{i,j} \leq E_{i,j} \forall (i,j) \in E \\
&\sum_{(i,j) \in \mathcal{C}} y_{i,j} \leq |\mathcal{C}| - 1 \forall \mathcal{C} \\
&x_i - y_{i,j} \geq 0 && \forall i, j \\
&x_j - y_{i,j} \geq 0 && \forall i, j \\
&z_{i,j,k} - y_{i,j} \leq 0 && \forall i, j, k \\
&z_{i,j,k} - y_{j,k} \leq 0 && \forall i, j, k \\
&z_{i,j,k} - y_{i,j} - y_{j,k} + 1 \geq 0 && \forall i, j, k \\
&x_i \in \{0, 1\} \\
&y_{i,j} \in \{0, 1\} \\
&z_{i,j,k} \in \{0, 1\}
\end{aligned} \tag{5.7}$$

5.5 Simulation

5.5.1 Network

We simulate our integer programming heuristic for the general follow-back problem on a Twitter friends network of three famous computer scientists, Yann LeCun, Andrew Ng, and Sebastian Thrun. We use the Twitter advanced programming interface to compile the collective "friends graph" which includes the three different users, the union of their friends, and the friend/follower connections between all. The properties of the graph are in 5.1. We use the same simulation methodology delineated in 4.2.1.

Table 5.1: Computer Scientist Friends Graph

Friends graph	@ylecun, @SebastianThrun, @AndrewYNg
Number of targets	3
Number of vertices	218
Number of edges	3140
Density	0.07
Mean degree	14.40
Max out degree	66
Min out degree	1
Max in degree	113
Min in degree	0
Number weak components	1
Largest component diameter	6
Sum of vertex susceptibilities	46.77

5.5.2 Policies

We use the linearized integer program 5.7 to solve for the optimal policies. For this simulation, we set $c = 20$, implying that the agent is allowed to interact with at most twenty users in the friends' graph. We derive one policy (Second Order) that optimizes over the first and second order paths in the objective but does not consider vertex or edge centralities. We derive another policy (First Order) that optimizes over the first order paths exclusively. We then experiment with different centrality scores C such as in-degree centrality, out-degree centrality, eigenvector centrality, edge-betweenness centrality (in which case we modify our objective, replacing the term $c_i x_i$ with $c_{i,j} y_{i,j}$, where $c_{i,j}$ is the edge betweenness score between i and j and $y_{i,j}$ is still the binary edge variable associated with (i, j)). We normalize each centrality score so that they sum to one. For each vertex centrality measure, we also derive a policy that excludes the second order term from the objective and simulate with the resulting interaction sequences. So, for example, the policy "Eigenvector First-Order" selected a partial order and optimized over the vertex weights of the targets, the eigenvector centrality regularizer and the first order paths of the objective function alone.

We compare the optimization-derived policies with two random policies: 1) a policy (Random append) that selects a random permutation of seventeen members of the friends graph and then appends a random permutation of the three target users

and 2) another policy (Random) that realizes a random permutation of twenty users in the graph that must include the three targets. Note that the different between the two random policies is that the first one constrain the three targets to be at the end of the sequence, whereas the second version allows the three targets to fall anywhere in the interaction sequence. In addition, we compare the integer programming policies with policies that are based solely on network centrality measures. For example the policy"Eigenvector Centrality" takes the input "friends graph", removes the three targets, orders the seventeen vertices with the greatest eigenvector centrality in decreasing order of their scores. Then, calculates the eigenvector centralities on the three-vertex graph with just the targets, orders the targets by decreasing eigenvector centrality score, and appends this sequnce to the already ordered list of seventeen friends to make a sequence of twenty vertices. The results are listed in 5.2. We simulate each policy 10,000 times and show the fraction of times that each user followed back as well as the sum of the three fractions precentages. For each user, the maximum possible value is 1 and the maximum possible value of the sum is 3. The a priori follow probabilities for the three users are listed in the first row of 5.2.

As we would expect, the random policy that constrains the targets to fall at the end of the sequence outperforms the random policy that allows that targets to appear anywhere. Since the follow probabilities are increasing in overlap, we maximize the follow probability of a user by being followed by his friends first. Most of the policies that are based solely on centrality measures perform similarly to the random policies. Of note, the policies based on in-degree centrality and eigenvector centrality perform better than the other centrality-based policies. Eigenvector centrality performs 20% better than the next best centrality-based policy. Since eigenvector centrality weights the importance of its nodes by the importance of its neighbors [7], we intuitively understand that ordering interactions based upon this measure might help to capture valuable paths in the graph that helps increase the overlap phenomenon.

For each centrality measure, both first-order and second-order integer programming policies outperform the purely centrality-based policy. This is not surprising since the integer programming policies (IP) take into account the a priori vertex

Table 5.2: Computer Scientist Friends Graph Results

Policy	@ylecun	@SebastianThrun	@AndrewYNg	Sum
Prior	0.143	0.083	0.110	0.336
Random	0.177	0.090	0.144	0.410
Random Append	0.260	0.087	0.156	0.503
Eigenvector Centrality	0.378	0.084	0.169	0.630
Eigenvector First-Order	0.443	0.087	0.151	0.682
Eigenvector Second-Order	0.407	0.082	0.155	0.643
In-Degree Centrality	0.263	0.086	0.186	0.535
In-Degree First-Order	0.421	0.084	0.165	0.670
In-Degree Second-Order	0.422	0.088	0.138	0.647
Closeness Centrality	0.164	0.086	0.139	0.389
Closeness First-Order	0.460	0.082	0.156	0.692
Closeness Second-Order	0.452	0.081	0.147	0.680
Out-Degree Centrality	0.177	0.090	0.144	0.410
Out-Degree First Order	0.470	0.084	0.132	0.687
Out-Degree Second Order	0.425	0.083	0.148	0.647
Load Centrality	0.205	0.078	0.156	0.439
Load First-Order	0.370	0.087	0.156	0.614
Load Second Order	0.373	0.079	0.155	0.607
Edge Betweenness	0.4417	0.085	0.149	0.675
Second Order	0.442	0.086	0.143	0.670
First Order	0.447	0.083	0.140	0.670

susceptibilities whereas the centrality policies do not. The IP policies bias towards selecting vertices that have high prior follow probabilities. Some of the IP poliies that optimize over the centrality regularizer outperform the IP that optimizes over first and second order paths exclusively while others do not. Specifically the eigenvector, out-degree, in-degree, and edge betweenness regularizers perform better than the path-only policy. Centrality regularizers can only add value to the integer program if the correct centrality measure is selected.

There is little difference between the performance of the first order IP policies and the second order IP policies. The first order policy with no centrality regularization performed virtually the same as its second order counterpart. In fact, for all the centrality regularizers, the first order policies outperform the integer programs that include second order terms. We believe this might have taken place because our approximation for calculating expected follows assumes that the follow probability is linear in overlap, which may not be the case. Thus, the expression might be overcounting the impact that second order terms have on the objective value. When added to the first order objective, vertex centrality regularizers improved the policy's performance by as much as 2%.

These findings imply that including second order terms in the optimization may not be essential and using an appropriately selected network centrality regularizer could serve as a valuable surrogate for the higher order terms of the problem. This result is particularly valuable when using integer programming to solve for follow-back policies on large scale graphs. By removing the second-order term from the objective, we can reduce the size of our problem from N^3 binary variables to N^2, significantly increasing tractability.

Chapter 6

Future Work and Conclusions

6.1 Future Work

After determining an interaction sequence, an agent would want to know how to time his interactions. For instance, how long after interacting with the first user in the sequence should the agent wait until he interacts with the second user, and so on? Should the agent wait until the first user decides to follow him? What if this never occurs? The agent should not be expected to wait indefinitely. What if there is a cost per unit time that is incurred while the targets are not following the agent? If the agent does not gain the targets as followers after some period of time, perhaps the targets decide to follow a competitor. The longer the agent takes to acquire the targets as followers, the greater the chance that they are acquired by a competitor. If the agent is aware that a user will take too long to follow-back, he might decide not to waste time interacting with that user.

6.1.1 Timing Experiment

To incorporate timing into our optimization, we must first understand if and what features of a Twitter user's behavior is indicative of how quickly he will follow-back. We conducted one final experiment with the same agents we used in 2.

We had each agent interact with users that had tweeted about "Morocco" or "arts

Table 6.1: Follow-back Delay

Feature	β	p-value
Intercept	2.38	0.07
Number of tweets	-4.97	0.67
Percentage retweets	8.26	0.62
Percentage hashtag	6.01	0.73
Percentage media	6.39	0.70

The significance codes are *** : 0.001, ** : 0.01, and * : 0.05.

and crafts" within the past twenty-four hours. Then, we had one of our six agents interact with these users, where an interaction was a retweet and a follow. After interacting with each user, we had the agent check its followers list each minute to see if and when the user it interacted with followed the agent.

Over a 1.5 week period, the agents interacted with 500 Twitter users. Of the 500 interactions, 64 users followed the agents back. The 13% follow-back rate is consistent with what we observed in 2. For each user that followed-back, we gathered some of the features of that user's Twitter behavior such a as the number of tweets they have posted in total, the percentage of these tweets that are retweets, the percentage that contain pieces of media such as linked photos are videos, and the percentage that contain hashtags. We then used linear regression to predict that amount of time it took the users to follow-back, regressing on these features. The results of the regression are shown in 6.1.

None of the features we checked were significant in predicting the delay in how long it takes a user to follow-back. However, we believe that given enough data and the proper experimental set up, some user features may be significant in predicting the follow-back delay. We conjecture that people who use Twitter more frequently are more inclined to follow-back faster, as they will likely see the agent's interaction sooner. A future line of research could involve finding viable features.

Modeling time delays between interactions would also increase the complexity of our general follow-back dynamic programming problem presented in 1.5. Future research could use the framework of continuous-time dynamic programming to analyze the problem and gain insight. Incorporating timing decisions into the integer program

outlined in 5 would greatly increase its complexity and could potentially become an issue. This is another area for future research.

6.1.2 Maintaining Connections

The follow-back problem focused on establishing a connection with a primary target. However, once a connection is established, it can be strengthened through other types of interactions such as replying or retweeting. A natural next step in this work is to develop optimal policies for maintaining or strengthening connections in a social network. Further empirical analysis will be needed to understand the dynamics of repeated interactions. However, once a model for the effect of such interactions on a social connection is developed, the follow-back problem can be extended to find optimal policies for maintaining or strengthening connections through repeated interactions in a social network.

6.2 Conclusion

We have proposed the follow-back problem for targeting users in a social network. We conducted empirical studies in Twitter to find what user features and interaction types cause users to follow. Specifically, we identified the importance of a agent's local "overlap" with a target user in gaining the target's follow. Using this analysis we construct a simple model for the follow probability. We then use this model to develop optimal policies for the follow-back problem. Through this we develop the notion of a follow-back score on a directed acyclic graph.

We come up with a graph theoretic algorithm to derive a follow-back policy on an arbitrary graph which we see perform well compared with other policies in simulation. We also model our problem as a constrained integer program to select an optimal directed acyclic subgraph of an arbitrary directed graph.

We truncated some of the higher order terms to make the problem tractable on large scale Twitter networks. We used network centrality regularizer functions to replace some granularity lost in the objective by truncating those higher order terms.

We show in simulation that this approach performs well. Our work supports the power of network centrality measures. When combinatorial optimization in graphs becomes intractable, network centrality measures serve as useful functions that are cheap to compute. Researchers may, at times, find it viable to discard intractable portions of their math programs and use network centrality regularizers as helpful surrogates.

6.3 Application

Learning how to network with others and spread information has uses from sectors of society ranging from defense to business. People have long understood the skills needed to network and communicate effectively in the offline setting and have recognized the importance of these skills. As the adoption of social media has made networking and spreading information easier, organizations that want to remain competitive must begin using more strategic ways of connecting with others online. If they do not, they risk being drowned out in the ever increasing volume of content that travels around cyberspace. Though the follow-back problem was framed over Twitter, the intuition and modeling can be used to more effectively connect with users in other types of social media networks. Our model is merely preliminary work in a rapidly growing field.

Bibliography

[1] Joint Publication 3-13.2. *Military Information Support Operations*. Joint Chiefs of Staff, 2010.

[2] H Schichl A Baharev and A Neumaier. An exact method for the minimum feedback arc set problem. *University of Vienna*, 2015.

[3] Jill R. Aitoro. What can dod chief ash carter learn from facebook's sheryl sandberg. *Washington Business Journal*, 2015.

[4] Sinan Aral and Dylan Walker. Identifying social influence in networks using randomized experiments. *IEEE Intelligent Systems*, 26:91–96, 2011.

[5] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 337:337–341, 2012.

[6] J Berger and Jonathan Morgan. The ISIS Twitter census: Defining and describing the population of ISIS supporters on twitter. *The Brookings Project on US Relations with the Islamic World*, 3:20, 2015.

[7] P. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.

[8] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. Design and analysis of a social botnet. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 57:556–578, 2013.

[9] Rukmini Callimachi. Isis and the lonely young american. *The New York Times*, 2015.

[10] Damon Centola. The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197, 2010.

[11] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in Twitter: the million follower fallacy. In *Proc. AAAI Conf. on Weblogs and Social Media*, 2010.

[12] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[13] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[14] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[15] Nicholas A Christakis and James H Fowler. The collective dynamics of smoking in a large social network. *New England journal of medicine*, 358(21):2249–2258, 2008.

[16] J. Kleinberg D.M. Romero. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *4th Int'l AAAI Conference on Weblogs and Social Media*, 2010.

[17] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets*. Cambridge University Press, 2010.

[18] L. Singhi F. Nagle. Can friends be trusted? exploring privacy in online social networks. In *IEEE International Conference on Advanaces in Social Networking Analysis and Mining*, pages 312–315, 2009.

[19] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

[20] L. C. Freeman, S. Borgatti, and D. R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13:141–154, 1991.

[21] J Pan G King and M Roberts. How censorship in china allows government criticism but silences collective expression. *American Political Science Review*, 2013.

[22] C Marlow J Ugander, L Backstrom and J Kleinberg. Structural diversity in social contagion. *Proceedings of National Academy of Sciences*, 109:5962 – 5866, 2012.

[23] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[24] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[25] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Automata, languages and programming*, pages 1127–1138. Springer, 2005.

[26] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604 – 632, 1999.

[27] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. WWW*, 2010.

[28] A. Smith M. Duggan. Social media update 2013. http://www.pewinternet.org/2013/12/30/social-media-update-2013/, December 2013.

[29] USAF Major Christopher J. McCarthy. Anti-access/area denial: The evolution of modern warfare. *US Naval War College*, 2013.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University InfoLab, 1999.

[31] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581–603, 1966.

[32] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who's the culprit? *Information Theory, IEEE Transactions on*, 57(8):5163–5181, 2011.

[33] Richard P. Stanley. *Enumerative Combinatorics Volume I.* Cambridge University Press, 2012.

[34] Dustin Volz and Mark Hosenball. White house, silicon valley to hold summit on militants' social media use. *Reuters*, 2016.

[35] D. Younger. Minimum feedback arc sets for a directed graph. *Circuit Theory, IEEE Transactions*, 10:238–254, 1963.